

KU Leuven
Groep Biomedische Wetenschappen
Faculteit Geneeskunde
Departement Neurowetenschappen



FINGER MOVEMENT DECODING

FROM SOURCE LOCALISATION TO TENSOR REGRESSION MODELING

Axel Faes

Jury:

Promotor: Marc M. Van Hulle
Voorzitter leescommissie: Dirk Daelemans
Voorzitter openbare verdediging: Patrick Dupont
Juryleden:

Luc De Raedt
Dante Mantini
Alfred Meurs
Nick Ramsey

Proefschrift voorgedragen
tot het behalen van de
graad van Doctor in de
Biomedical Sciences
(PhD)

March 2023

KU Leuven
Biomedical Sciences Group
Faculty of Medicine
Department of Neurosciences

KU LEUVEN

DOCTORAL SCHOOL
BIOMEDICAL SCIENCES

FINGER MOVEMENT DECODING

FROM SOURCE LOCALISATION TO TENSOR REGRESSION MODELING

Axel Faes

Jury:

Supervisor: Marc M. Van Hulle
Chair examining committee: Dirk Daelemans
Chair public defence: Patrick Dupont
Jury members
Luc De Raedt
Dante Mantini
Alfred Meurs
Nick Ramsey

Dissertation presented in
partial fulfilment of the
requirements for the
degree of Doctor in
Biomedical Sciences
(PhD)

March 2023

Preface

This thesis has been quite an adventure, sailing through uncharted waters. Luckily, I had a compass in the form of my mentor and supervisor, Marc Van Hulle.

First of all, I would like to express my sincere gratitude to prof. Van Hulle. When I took his course, Brain-Computer Interfaces, as a Masters student, I was instantly enamored with the field. Thank you for welcoming me in the lab five years ago and allowing me to conduct this research, for the confidence and the freedom you gave me to accomplish this work.

My sincere gratitude to all members of the jury for agreeing to read and help me to improve the manuscript, and to participate in the defense of this thesis. I am grateful to prof. Mantini and prof. De Raedt for his willingness to meet and spend much time to provide me with their much appreciated advice. I would also like to thank the external jury members, prof. Meurs and prof. Ramsey for agreeing to be part of the examining committee and for sharing their expertise and insights during the reviewing of this thesis. After reading your great works, I was honored to make your acquaintance and to share my findings with you.

I want to thank my current and former colleagues with whom I enjoyed the chats, lunches, the funny and stressful moments. I extend my thanks also to all members of the Department of Neurosciences, who contributed - even with a smile when passing in the corridor - to make working in the lab an even better experience. My deepest thanks to my dad, mom and godfather for their constant and unconditional support thought the years, both in my study and my life in general. I would not be here if it were not because of you.

Finally, I want to thank all my friends for your showing sincere interest in what I have been working on over the years. I would also like to thank my sportscoach, Aron, for helping me to get into shape and making my life outside research so much more healthy.

Abstract

Brain-Computer Interfaces (BCIs) are hailed for bypassing defective neural pathways by translating brain activity directly into actions that convey the user's intent. How the kinematics of muscular activity relates to the motor- and somatosensory activity in the brain has been the focus of recent advancements. With such motor BCIs, amputees are able to gain control over a prosthesis and stroke patients to regain control over a paralyzed limb via electrical stimulation of their dysfunctional muscles or via an exoskeleton that supports the intended movements. The superior spatio-temporal resolution, bandwidth, and recording stability of electrocorticography (ECoG), a partially invasive brain recording technique, yields a new outlook on motor BCI applications. Despite some stunning successes in arm- and hand movement control from ECoG, the precise decoding of finger movements, which is essential for daily activities, is still lacking. A possible reason is that current decoders rely on conventional one- or two-way regression models, which might not adequately capture the intricate relation between neural activity and intended and unintended (such as coactivations) finger movements. The main objective of this PhD is to develop a robust, accurate, and quick-to-train decoder that predicts single- and coordinated finger trajectories from ECoG recordings. We used multiway decoders as they preserve the multilinear structure of the data while taking advantage of potentially hidden multilinear components. We demonstrated cutting-edge performance with the proposed decoders. As multiway models tend to be slow to train, which may become a significant obstacle for their clinical adoption, we also investigated whether the proposed multiway decoders could be used in a real-time setting. The findings support the relevance of the proposed multiway decoders for real-time ECoG-based finger activity, providing in this way an outlook on achieving hand dexterity.

Beknopte samenvatting

Brain Computer Interfaces (BCI) registeren hersenactiviteit om ze naar intentionele acties te vertalen en omzeilen zo de behoefte aan spieractiviteit. Recente ontwikkelingen spelen in op de relatie tussen de kinematica van menselijke bewegingen en gelokaliseerde activiteit in motorische en somatosensorische hersengebieden. Motor BCIs worden gepromoot als oplossing voor verlamde patiënten door hen in staat te stellen een robotledemaat aan te sturen of opnieuw de controle over het verlamde ledemaat te verwerven via de electrische stimulatie van de gedeconecteerde spieren of via een exoskeleton dat de intentionele bewegingen ondersteunt. De hoge spatio-temporale resolutie, de brede bandbreedte, en de lange-termijn stabiliteit van elektrocorticografie (ECoG), een deels niet-invasieve registrati 技术, biedt nieuwe perspectieven voor motor BCIs. De precieze decodering van autonome individuele vingerbewegingen, wat essentieel is voor de dagelijkse activiteiten van de beoogde patiëntengroep, ontbreekt nog steeds ondanks enkele verbluffende successen van motor BCIs in nauwkeurige arm- en handcontrole uit ECoG-signalen. Doch de huidige decoders zijn gebaseerd op conventionele één- of tweerichtingsregressiemodellen die de ingewikkelde relatie tussen de neurale activiteit en de beoogde bewegingen van de vingers niet adequaat capteren. Er is een duidelijke behoefte aan geavanceerde decoderingsalgoritmen om vingervaardigheid aan de bedoelde patiënten te kunnen bieden. Het hoofddoel van dit doctoraat is het creëren van een decoder die intentionele vingerbewegingen uit ECoG registraties haalt die bovendien snel te trainen, robuust en nauwkeurig is. We betrouwen op de multiway-benadering omdat deze het potentieel heeft om verborgen multilineaire componenten te exploiteren terwijl de multilineaire structuur van de gegevens behouden blijft. Meer specifiek hebben we gekeken naar het vermogen van multiway-modellen (decoders) om het traject van individuele vingers te controleren met behulp van ECoG registraties en hebben we een klasse van multiway-modellen geïdentificeerd die tot continue regressie en (asynchrone) classificatie in staat zijn. We hebben aangetoond dat de voorgestelde decoders superieur zijn aan geavanceerde lineaire en multiway-modellen. We hebben ook gekeken of de voorgestelde

benaderingen konden voldoen aan de eisen van real-time operatie, omdat de tijdcomplexiteit van multiway modellen een aanzienlijk obstakel zou kunnen zijn voor hun acceptatie. Over het algemeen ondersteunen de bevindingen het gebruik van multiway-decoders voor ECoG-gebaseerde vingeractiviteit in real-time, waardoor de klinisch belangrijke vingervaardigheidstoepassingen in het vizier komen.

Contents

Abstract	iii
Beknopte samenvatting	v
Contents	vii
List of Figures	xi
List of Tables	xvii
1 Introduction	1
1.1 Human Motor System	1
1.2 Brain Computer Interfacing	1
1.3 Motor Movement Decoding from EEG	4
1.4 Multiway Finger Movement Decoding from ECoG	7
1.5 Problem Statement	7
1.6 Proposed approach	8
1.6.1 Multiway linear regression: achievements and limitations of the BTTR model	8
1.6.2 organisation	10
2 Source Space Reduction for eLORETA	19
2.1 Abstract	19
2.2 Introduction	20
2.3 Materials and Methods	22
2.3.1 Simulated EEG Dataset	23
2.3.2 Real-world EEG Datasets	25
2.4 Results	26
2.4.1 Simulated EEG Dataset Results	26
2.4.2 Clustering reconstructed dipoles	28
2.4.3 Real-world EEG Dataset Results	29

2.5	Discussion	31
2.6	Conclusion	34
2.7	Funding	35
3	Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data	39
3.1	Abstract	39
3.2	Introduction	40
3.3	Materials and Methods	42
3.3.1	Simulation Procedure	42
3.3.2	Connectivity Models	47
3.3.3	Performance Evaluation	53
3.4	Results	53
3.4.1	Ground Truth 1	53
3.4.2	Ground Truth 2	60
3.5	Discussion	62
3.6	Conclusions	66
3.7	Acknowledgement	66
3.7.1	Author Contributions	66
3.7.2	Funding	66
3.7.3	Conflicts of Interest	67
4	Single Finger Trajectory Prediction From Intracranial Brain Activity Using Block-Term Tensor Regression With Fast and Automatic Component Extraction	71
4.1	Abstract	71
4.2	Introduction	72
4.3	Previous work on automatic multilinear tensor rank selection	75
4.4	Methods	76
4.4.1	Component extraction and selection	76
4.4.2	Block-Term Tensor Regression (BTTR)	81
4.4.3	Time Complexity Analysis	83
4.4.4	Non-multilinear approaches	83
4.5	Data Sets Used	84
4.5.1	Synthetic Data For Component Analysis	84
4.5.2	Regression Analysis Of Real-World Data: Decoding Finger Movement Trajectories From ECoG Signals Recorded in Humans	86
4.6	Results	87
4.6.1	ACE And ACCoS On Synthetic Data For Component Extraction	87
4.6.2	BTTR on Real-World Data: Decoding Finger Movement Trajectories from Human ECoG Recordings	89

4.7 Conclusion	92
5 Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression	101
5.1 Abstract	101
5.2 Introduction	102
5.3 Materials and Methods	104
5.3.1 Extended Block-Term Tensor Regression (eBTTR)	105
5.3.2 Non-multilinear approaches	109
5.3.3 Dataset	110
5.3.4 Parameter optimization multilinear models, performance assessment	113
5.4 Results	113
5.5 Conclusion	114
5.6 Funding	115
6 Conclusion	121
6.1 General Conclusion	121
6.2 Future Research	123
6.2.1 Sign Language Alphabet Decoding	123
6.2.2 Model training based on observed movement	124
6.2.3 Subject Training	124
6.2.4 Finger Abduction Decoding	125
A Source Space Reduction for eLORETA	131
A.1 Simulated EEG Dataset	131
A.1.1 Initialisation	131
A.1.2 Methods	131
A.1.3 Medoid Clustering	134
A.2 Imaginary Movement EEG Dataset	134
A.2.1 Head Model	134
A.2.2 Methods	135
A.3 Cuban EEG Dataset	135
A.3.1 Methods	135
B Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data	139
B.1 Supplementary Material	139
C Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression	141
C.1 Supplementary Material	141

Curriculum Vitae	143
Publications	155
D Acknowledgements	156
E Personal Contribution	157
F Conflict of Interest	158

List of Figures

1.1	Motor Homunculus and its disposition over the primary motor cortex. Source: PhysioOsteoBook.	2
1.2	BCI diagram.	3
1.3	Left panel: Schematic rendition of an ECoG grid implant. M= motor cortex, S= somatosensory cortex. Middle panel: Text-spelling application controlled by a late-stage ALS patient with an ECoG strip implant covering the right hand brain area [4]. Right panel: exoskeleton controlled by a tetraplegic patient [12].	4
1.4	EEG Source Localisation [21].	5
1.5	Evidence of nonlinear ECoG responses in high gamma band (γ_3 , see further). Top row: When performing fast index finger repetitions (2Hz) (right panel), the second γ_3 peak is lower than the first one (left panel). Bottom row: When performing slow repetitions (1Hz, right panel), the peaks are similar (left panel). Thick lines represent averages over 30 trials, shaded areas standard deviations. Spectral results for 5 Hz sampling rate. Data set from [52]	9
1.6	Left Panel: Digital data glove. Right Panel: Hand exoskeleton.	12
2.1	Average correlation (blue line, left scale) and complement of the average normalized spatial spread (red line, right scale), normalized with respect to eLORETA's outcome (red ball), plotted as a function of the sparsity constraint κ . The blue ball denotes eLORETA's average correlation. The Pareto-optimal solution is at the intersection ($\kappa \approx 0.08$).	27

2.2	Distribution of reconstructed dipoles (brain top views, nose pointing upwards) of the simulation study, (uncoherent) pink noise case. Shown are the results obtained using eLORETA, Sparse eLORETA, BC-VARETA and LCMV given the Far Superficial configuration. with the 3 medoid clusters marked in orange, green and cyan. Medoids centers are indicated with black filled circles, the ground truth dipoles with red filled circles. In the bottom right cluster (green for eLORETA), the medoid center overlaps with the ground truth dipole.	30
2.3	Distribution of reconstructed dipoles (brain top views, nose pointing upwards) of the Cuban EEG dataset using eLORETA, Sparse eLORETA, BC-VARETA and LCMV. The grey nodes in the background are mesh nodes. The filled circles are active dipoles with a darker color indicating more activity.	31
2.4	ERD/ERS post-cue imagined right hand movement activation of 300 sources (aka bootstrapping) obtained with eLORETA, Sparse eLORETA and BC-VARETA, mapped on left and right lateral brain views (arranged column-wise). The grey nodes in the background are mesh nodes. Blue dots correspond to ERS, yellow/red to ERD and green to neither. For BC-VARETA dipoles are grey dots as no ERS/ERS assessment is possible. . .	32
2.5	Contralateral view of the Sparse eLORETA solution with clusters labeled by the corresponding brain regions.	33
3.1	Simulation Procedure followed by Connectivity Estimation. TCDF = Depthwise Separable 1D Temporal Causal Discovery Framework, LSTM-NUE = Long Short-Term Memory with Non-Uniform Embedding, Conv2D = 2D Convolutional Network, TRGC = Time-Reversed Granger Causality. Coordinates in the Ground Truths denote MNI-coordinates.	43
3.2	Ground Truth 1 with three fixed dipoles.	44
3.3	Ground Truth 2 with two fixed, one moving dipole.	45
3.4	Locations (MNI-coordinates) of the dipoles the Ground Truths have in common. Left: Far-Superficial fixed dipoles. Right: Far-Deep fixed dipoles.	45
3.5	CV with length of first train-fold = length of first test-fold (= 1500/7).	51
3.6	Main evaluation measures.	53

3.7	Mean Sensitivity for all methods ($L = 1500$). Sensitivity Ranking Far-Superficial: TRGC > Conv2D (TS = 2) > LSTM-NUE > TCDF > Conv2D (TS = 3). Sensitivity Ranking Far-Deep: TRGC > LSTM-NUE > Conv2D (TS = 2) > Conv2D (TS = 3) = TCDF. Abbreviation TS = amount of time series included in the predictions.	54
3.8	Mean Precision for all methods ($L = 1500$). Sensitivity Ranking Far-Superficial: TRGC > Conv2D (TS = 2) > LSTM-NUE > TCDF > Conv2D (TS = 3). Sensitivity Ranking Far-Deep: TRGC > LSTM-NUE > Conv2D (TS = 2) > Conv2D (TS = 3) = TCDF. Abbreviation TS = amount of time series included in the predictions.	55
3.9	TCDF, Mean attention score rankings. Colors: “1”, green, denoting the highest attention score in one TCN—one column). The columns represent the targets (T X1, T X2, T X3), the rows the predictors (X1, X2, X3). Self-connectivity is excluded.	57
3.10	LSTM-NUE, Neural-Network Granger Scores rankings (Top) versus Truth 1 (Bottom), excluding self-connectivity. Color coding: dark green > light green > yellow > orange > red. The columns represent the targets, the rows the time series used for prediction.	58
3.11	Conv2D, R2-strength score for time series pairs (Top) versus ranking of connections in the Ground Truth 1 (Bottom, 1 being the strongest), including self-connectivity. Color coding: dark green > light green > yellow > orange > red. The columns represent the targets, the rows the time series pairs used for predicting the target time series.	59
3.12	Runtime for the training of all ANNs, with 3 time series (TS = 3), $L = 1500$. An extra comparison showing the runtime of Conv2D with two time series as predictors (TS = 2) is shown (but all datasets contain 3 time series). “Neurons” = the number of hidden layer neurons.	61
4.1	mPSTD	78
4.2	ACE	78
4.3	Overview of the proposed models for component extraction and selection.	79
4.4	Scheme of BTTR algorithm with 3rd-order predictor variable $\underline{\mathbf{X}}$ and 1st-order response variable y . Note that each block is computed using either ACE or ACCoS further referred to as ACE-BTTR and ACCoS-BTTR.	80
4.5	ACCoS	80
4.6	BTTR	82

4.7	Diagram with the steps to create an ECoG tensor.	87
4.8	Top left panel (Standard PSTD): C1 accuracy (color coded, see scale on the right) achieved by standard PSTD when (manually) varying model parameters SNR and τ . Top right panel (mPSTD): idem but for the proposed mPSTD. Bottom left (BIC-SNR) and right (BIC-Ratio for best SNR): filled circles indicate the selected SNR (50) and τ (99.7) values.	88
4.9	Results of correlation analysis when varying the correlation parameter α while keeping $R_2, R_3, R_4 = [2, 2, 2]$, $I_1 = 10000$, and $I_2, I_3, I_4 = [20, 20, 20]$. The reported values represent the correctness of the expected components (C1) and the extra components (C2 with average of number of extra components in brackets).	89
4.10	Predicted ring finger flexions of Subject 1. Actual (data glove) and predicted (regression models) finger flexion amplitudes (z-scores) are plotted as function of time (in ms), Curve colors are explained in the inset. For the sake of exposition, only results for the generalized version gBTTR are shown.	90
4.11	Performances of the proposed models and HPLS when predicting training and test data.	90
5.1	Scheme of eBTTR algorithm with 3rd-order predictor variable $\underline{\mathbf{X}}$ and 2nd-order response variable \mathbf{Y} . Note that each block is computed using ACE. In this work, $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples} \times \text{Channels} \times \text{Frequencies} \times \text{Time}}$ and $\mathbf{Y} \in \mathbb{R}^{\text{Samples} \times \text{Fingers}}$. . .	106
5.2	eBTTR	107
5.3	ACE	108
5.4	mPSTD	109
5.5	Unintentional coactivation for subject 3 during middle finger movement (visualized as left hand).	112
6.1	Contralateral view of the Sparse eLORETA solution with clusters labeled by the brain regions active during a motor task.	122
6.2	Left Panel: Digital data glove. Right Panel: Hand exoskeleton.	123
6.3	Excerpt of the Flemish sign language alphabet. An avatar is used to instruct the subject with the finger movements that lead to these signs, starting from a closed fist, to show the user what is decoded from his/her brain activity in real-time, or to have the attempts post-hoc evaluated by a sign language expert.	125

6.4	Outcome of our ECoG pilot study on performed sign language alphabet decoding using eBTTR. Left panel: the cued American sign language letters V and Y. Right panel: still images of the avatar of the decoded finger trajectories. The average Pearson correlations between the data glove- and decoded trajectories were above 70%.	126
6.5	Left Panel: Digital data glove. Right Panel: Hand exoskeleton.	126
A.1	Processing pipeline. Schematic rendition in the case of the simulated EEG dataset.	137

List of Tables

2.1	Fixed dipole coordinates within the subsampled New York head model, for each configuration. The first table lists the positions of the first fixed dipole, the second table those of the second fixed dipole.	24
2.2	Average correlation coefficients of the Far Superficial dipole configuration given 3 noise models. PN = (incoherent) pink noise, CPN = coherent pink noise, CAN = autoregressive pink noise.	26
2.3	Average correlation coefficients in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.	26
2.4	Average elapsed time (in seconds) of LCMV, eLORETA, Sparse eLORETA and BC-VARETA, determined using Matlab's tic toc function, averaged over 1000 runs.	28
2.5	Average spatial spread in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.	29
2.6	Average distance between medoids and true sources in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.	29
3.1	Set-up of the compared ANNs and TRGC.	47
3.2	Scores of the ANNs in comparison with TRGC, using Ground Truth 2. Results are based upon datasets where all 3 time series (TS) were included as predictors, with one exception: results from Conv2D with two time series as predictors, indicated with *, were also included.	61

3.3 Scores of the ANNs in comparison with TRGC, using Ground Truth 2. Results are based upon datasets where all 3 time series (TS) were included as predictors, with one exception: results from Conv2D with two time series as predictors, indicated with *, were also included.	62
4.1 Mathematical notation	77
4.2 Finger trajectory decoding performance (Pearson correlation) of BTTR and other models for Subject 1.	92
4.3 Idem to Table 4.2 but for Subject 2.	93
4.4 Idem to Table 4.2 but for Subject 3.	93
5.1 Mathematical notation	105
5.2 Intended (cued) finger movement accuracy for Subject 1 (Pearson correlation).	114
5.3 Idem to Table 5.2 but for Subject 2.	114
5.4 Idem to Table 5.2 but for Subject 3.	115
B.1 Scheirer–Ray–Hare Test using Sensitivity as a criterion. Alpha level = 0.05. Significant results shown in bold. Models: Conv2D, TS = 2, kernel size 4×2 versus 4×4 , Conv2D, TS = 3, kernel size 4×3 versus 4×4 , with TS denoting the amount of time series. Model ($H(3,32) = 16.04$, $p = 0.001$) and dipole condition ($H(1,32) = 4.53$, $p = 0.03$) have significant effects on Sensitivity. Post hoc Mann–Whitney U tests (Bonferroni-corrected alpha level = 0.008) revealed that Sensitivity was significantly higher for Conv2D, TS = 2 (Mdn = 0.67, 0.67), versus Conv2D, TS = 3 (Mdn = 0.00, 0.17), $p = 0.006$, $p = 0.007$ for differing kernel sizes.	140
C.1 Row-wise: Subject 1's unintended finger movement accuracies for for eBTTR / BTTR; between brackets the ratio between the unintended and intended (cued) finger flexions; the cued finger is marked by “-”, the accuracy listed in Table 2 in the results section. The finger flexion ratios show to what extent the non-cued finger moves along with the cued finger. Note the difference between thumb and index finger and the significant difference in accuracy between eBTTR and BTTR's predicted unintended movements. Accuracies are expressed by the Pearson correlation coefficient between predicted and actual finger movement trajectories. . .	142
C.2 . Idem to Table C.1 but for Subject 2.	142
C.3 Idem to Table C.1 but for Subject 3.	142

Chapter 1

Introduction

1.1 Human Motor System

Motor actions result from multistage cortical processes of which the motor system controlling the muscles is the front-end. There is an interplay with visual information processing when we chart the surroundings to locate an object in our peripersonal space we decide reach out to, and to explore possible grasp options, or to walk around it when obstructing our path. The main drivers of motor actions come from cortical regions that reside in the frontal lobe: the prefrontal-, premotor-, and primary motor cortex. These regions plan and issue the motor commands, possibly in a coordinated manner when fingers are sequentially activated to grasp a complex object. The prefrontal cortex specifies the general goal toward which movements should be directed, while the premotor cortex plans the sequence of movements needed to accomplish this goal and the primary motor cortex issues the commands that travel through the spinal cord to reach the targeted muscles. The primary motor cortex contains a map of the muscles of the body with clusters of neurons controlling specific muscles (see Motor Homunculus, Fig. 1.1). For instance, the execution of finger movements originates from neural activity of a cluster associated with the hand area.

1.2 Brain Computer Interfacing

Every year, half a million patients worldwide face paralysis due to the disruption of signal pathways between the brain and the muscles primarily caused by a spinal cord injury, brainstem stroke or amyotrophic lateral sclerosis (ALS) [1].

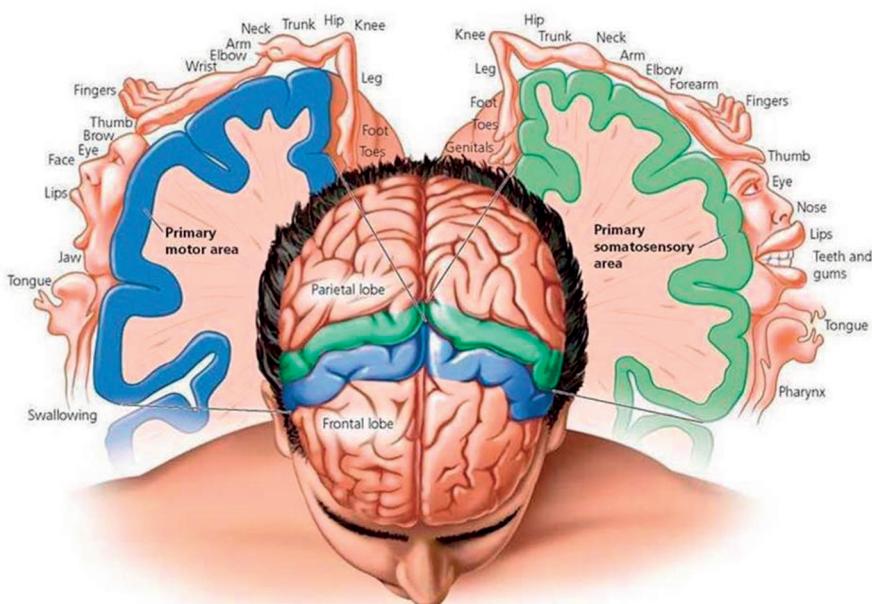


Figure 1.1: Motor Homunculus and its disposition over the primary motor cortex. Source: PhysioOsteoBook.

Various assistive technologies have been proposed to provide an alternative means of communication (for a review see [2]), including brain computer interfaces (BCIs) used, among others, to select letters one-by-one to spell out words.

A BCI is a computer-based system that acquires brain signals, analyzes and translates them into commands that are relayed to an actuator carrying out the desired action. Thus, BCIs do not use the brain's normal output pathways of peripheral nerves and muscles. As it is based on signals produced by the central nervous system (CNS) only, it differs from other assistive devices such as a voice-activated or muscle-activated communication system.

BCIs could help people with severe motor disabilities to interact with their environment, communicate, and enhance independence. BCIs have been shown to provide control over external devices such as wheelchairs, text spellers, domotics, and even speech synthesizers. Motor BCIs specifically aim to record motor cortical activity when bypassing disconnected neural pathways. Hereto motor cortical signals are recorded, distinctive features extracted and the results translated ("decoded") into actions. Recent advances include the decoding of 2D imagined index finger movements to control a Dasher typing interface

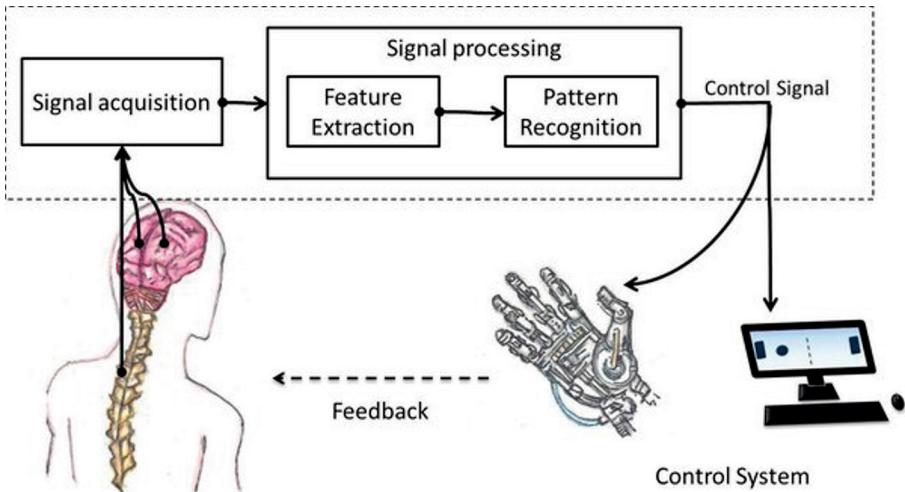


Figure 1.2: BCI diagram.

(BrainGate2 system, [3]) and imagined hand closing to select letters [4] (Text-spelling application, Fig. 1.3, middle panel). Brain motor cortical signals have also been translated into motor commands directly by functional electrical stimulation (FES) of the paralyzed hand muscles [5], [6] or by controlling prosthetic hands or arms [7]–[11] and exoskeletons [12] (Exoskeleton, Fig. 1.3, right panel).

Motor BCIs primarily rely on wire microelectrodes implanted in the cortical tissue to provide recordings with high temporal and spatial resolution – even down to individual neurons – enabling motor control with many degrees of freedom. However, besides causing irreversible damage to the brain tissue and its vasculature, the implant could also trigger fibrous scar tissue formation that eventually affects long-term signal stability [13]. Over the last decade, electrocorticography (ECoG) has been gaining attention in the BCI research community as it can successfully address the long-term stability issue [14] while providing high temporal, spatial and spectral resolution. ECoG uses electrodes placed on the exposed cortical surface to record electrical activity (Fig. 1.3, left panel). It thus prevents damage to the cortical tissue and its vasculature and avoids both scarring and other histological processes [15]. When recorded from the primary motor cortex, ECoG signals exhibit significant motor-related spatio-temporo-spectral patterns that can be exploited to capture the dynamics of the associated movement [16]–[18].

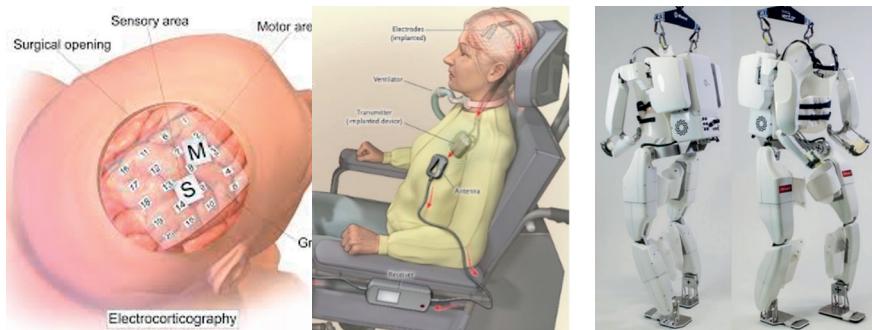


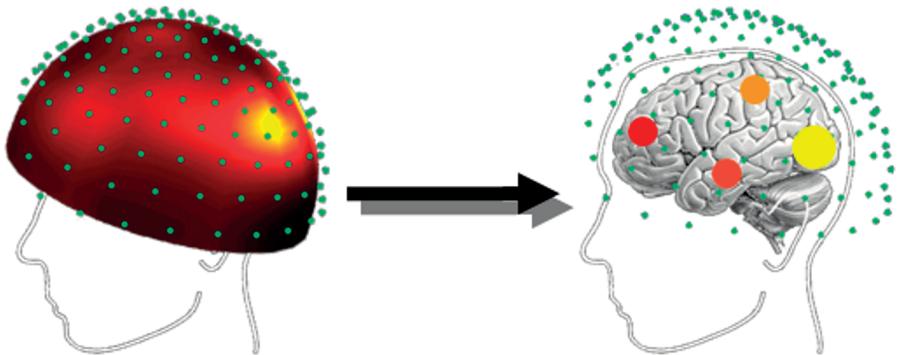
Figure 1.3: Left panel: Schematic rendition of an ECoG grid implant. M= motor cortex, S= somatosensory cortex. Middle panel: Text-spelling application controlled by a late-stage ALS patient with an ECoG strip implant covering the right hand brain area [4]. Right panel: exoskeleton controlled by a tetraplegic patient [12].

1.3 Motor Movement Decoding from EEG

Electroencephalography (EEG) is the most widely used BCI modality to record brain signals. Motor imagery (MI) BCIs, which draw on imagined muscular activity, hold promise for stroke rehabilitation, controlling assistive devices such as wheelchairs, text spellers, and other applications for paralyzed individuals. Due to the broad scalp coverage of EEG, it is possible to tap into brain networks. However, locating the brain regions (“sources”) that make up that network and extracting their connectivities can be problematic due to the high signal correlation between neighboring electrodes as a result of the volume conductance effect as brain activity is spreading out before reaching the electrodes. Volume conduction is the mixing and attenuation of different sources due to the low conductivity of the layers that separate the cortex from the surface of the scalp. Therefore, meaningful connectivity patterns can be estimated only after localizing the brain sources, a topic generally referred to as source localization, and extracting their activities. Still, spurious connectivities may persist in source-reconstructed EEG, rendering it vital to choose an appropriate measure of connectivity. Understanding brain connectivity aids our knowledge of the working of the brain. It allows us to better understand sensorimotor and cognitive tasks that are performed. This can lead to improved diagnosing of various diseases such as aphasia and epilepsy par excellence (i.e., epileptogenic zone localization prior to resective surgery) [19]. With regards to finger movement decoding, only movement/no-movement classification is a realistic goal given the restricted spatial resolution of EEG (cf. volume conduction) and its limited signal-to-noise ratio (SNR) [17], [20].

Figure 1.4 visualizes the source localization problem (reverse problem), which is ill-posed as EEG electrodes are placed on the 2D scalp while the brain is 3D, and there are far less EEG electrodes than potentially active brain regions. Several constraints and assumptions have been proposed to relax ill-posedness but at the expense of yielding approximate solutions.

Figure 1.4: EEG Source Localisation [21].



Equation 1.1 formulates the source localisation problem with X the scalp-recorded EEG activity, S the electrical sources within the brain, the current density vector. L is the head volume conductor model, also called the Lead Field Matrix. The reverse problem is about finding S , which is represented in equation 1.2:

$$X = LS + n \quad (1.1)$$

$$O(S) = \min ||X - LS||^2 \quad (1.2)$$

Several head volume conductor models L have been proposed of which either the simple or realistic are preferred. Simple models assume the brain can be modeled by a sphere consisting of a few layers, making them fast to compute but at the expense of being not very accurate [22]. Realistic models are more accurate but computationally expensive, and rely on finite element or finite boundary techniques to construct them [23].

The electrical activity originating from a neural population of interest (“source”) can be modeled by a dipole S . The inverse problem involves determining the location, orientation, amplitude, and number of dipoles that contribute to the observed electrical activity. However, the number of dipoles that can be

considered is limited not the least by the limited spatial resolution of EEG due to volume conduction (cf. supra).

Algorithms such as dipole fitting, nonadaptive distributed-source imaging, and adaptive distributed-source imaging have been used to address the inverse problem. Dipole fitting involves calculating the locations, orientations, and magnitudes of a small number of dipoles. The dipole-fitting method can provide satisfactory results when a single dipole accounts for 80% of the observed electrical activity [24]. Dipole fitting can be performed by a number of different software programs or matlab toolboxes, including BESA, eeglab and fieldtrip [25]–[27]. Nonadaptive distributed-source imaging places thousands of dipoles within the brain on fixed locations; as the orientations is fixed, only the magnitude needs to be computed, which is done by assigning electrode weights to each dipole. As these electrode weights based on electrode locations, they are fixed over time and frequency. Note that sLORETA is a commonly used nonadaptive method [28]. Such methods are relatively fast to compute but allow for only a limited number of comparisons in statistical analyses [24]. Adaptive distributed-source imaging uses recorded data to compute the weights, and hence are not fixed over time and frequency. The accuracy of adaptive methods is often quite high, but at the expense of being more complicated to use given the many more parameters to fix. Beamforming is the most commonly used adaptive method [29]–[31].

In a simulated environment, a high spatial localization accuracy can be achieved, but this could become elusive when working with actual recordings. There are always uncertainties due to differences in electrode positions, brain anatomy and scalp conductivity, let alone artefacts arising from head and eye movement. This implies that the spatial accuracy is restricted to a few centimeters only [32]. The smallest voxels which are created by source reconstruction are typically 5-10 mm³ in size. [33]

Some source localization methods also provide neural activity predictions of the located sources, as in the case of linearly constrained minimum variance (LCMV) beamforming [30], a feature that was adopted in BCIs to feed the decoder with activity from specific regions of the motor cortex in an attempt to improve performance when controlled by imagined movements. Beamforming was also used by the host group to isolate EEG-ERP components of interest, even when their source location is not known [34], a concept that was further developed and integrated into several EEG-based BCI paradigms and applications [35]–[39]

.

1.4 Multiway Finger Movement Decoding from ECoG

When looking beyond movement/no-movement classification, decoding finger movement trajectories from ECoGs remains an unsolved problem, even though a few attempts have been reported. For example, Kubanek et al. [40] were the first to continuously decode performed flexions/extensions of individual fingers by analyzing local motor potentials (LMP) and spectral amplitudes in 5 frequency bands. The authors used a conventional sparse linear regression model and obtained an average decoding performance of $r= 0.52$ (predicted vs. expected Pearson correlation). Since then, continuous finger movement decoding has been studied based on gaussian- [41] and linear regression [42], [43], Convolutional Neural Networks (CNNs), Random Forests (RFs) and deep learning networks (proposed in [44]). However, finger trajectory decoding could benefit from recent developments in multilinear algebra (multiway decoding) as ECoG is in essence structured in space, time, and frequency domains. This structure is largely ignored in traditional vector- or matrix-based regression models, as the data is concatenated into vectors and matrices (aka unfolding). Multiway models preserve the multilinear structure of the data and support the discovery of potentially hidden multilinear components [45]. The two most popular multiway frameworks are Tucker (TKD) and CANDECOMP/PARAFAC (or CPD) decompositions (see [45], [46] for review) and both aim to determine the low-dimensional subspace where important information is residing.

In [47] two Block Term Decomposition (BTD) models were proposed to approximate N^{th} -order tensor data as a sum of K-blocks called rank- (L_1^k, \dots, L_N^k) BTD and rank- (L_1, \dots, L_N) BTD with $k=1, \dots, K$. The BTD approach has since then been adapted for regression analysis to model the relationship between arm trajectory and ECoG signals recorded from monkeys [48], [49] and recently between exoskeleton-based arm trajectory, arm- and wrist rotations and ECoG signals recorded from a tetraplegic patient [12]. Zhao et al. [48] developed a powerful generalized framework, called Higher-Order Partial Least Squares (HOPLS), based on $(1, L_2, \dots, L_N)$ -rank BTD (i.e. all blocks have the same multilinear rank), that provides enhanced predictability with optimal balance between fitness and model complexity.

1.5 Problem Statement

ECoG implantation is primarily used to locate the epileptogenic zone that need to be resected, but there has been an increasing number of studies using ECoG signals to develop BCIs. Steady progress is made toward neuroprosthetic limbs controlled via ECoG, but the journey has just begun and more research

needs to be done. One challenge that remains is a robust approach to decoding individual finger movements. Finger trajectory decoding could benefit from recent developments in multilinear algebra as ECoG is in essence structured in space, time, and frequency domains, which is neglected when relying on conventional methods that concatenate feature vectors into single (uniway) vectors. Cichocki et al. [45] advocated the use of high-order tensors (multiway arrays) in combination with multilinear techniques as the only way to discover potentially hidden feature dependencies in multiway data. The superiority of this approach has already been evidenced when decoding performed arm movement trajectories from ECoG in primates [50]. Despite encouraging results, multiway regression HOPLS is not suited for real-time applications because it often requires a prohibitive training effort.

State-of-the-art decoders such as HOPLS have returned superior results in terms of individual finger trajectory decoding, however, they are still inadequate when it comes to tackling more challenging everyday tasks, such as rapid finger repetitions (i.e. finger tapping), coordinated multi-finger movements (e.g. hand gestures) and sustained finger flexions (e.g. pointing or holding a cup). For example, my host lab found that during rapid finger repetitions gamma band ECoG activity decreases (1.5), and beta activity is suppressed, suggesting that such movements are facilitated. Another issue are sustained finger flexions where neural firing first increases when flexing a finger but then decreases when keeping it flexed [51] Such patterns are observed during both sustained flexing and resting.

1.6 Proposed approach

In this thesis, we investigate motor-related ECoG- and source-localized scalp electroencephalography (EEG) recordings. First, we perform source localization and extract connectivity patterns from motor-related EEG signals. Then, we propose a multiway Partial Least Squares framework called BTTR, as well as an extension of it, called eBTTR, for the accurate decoding of continuous finger trajectories while allowing for fast decoder (re-)calibration.

1.6.1 Multiway linear regression: achievements and limitations of the BTTR model

Our BTTR model adopts a deflation-based scheme that sequentially decomposes data into a series of blocks [53]. It offers increased modeling flexibility as it automatically computes a set of latent variables so that each block of the multiway predictor variable is maximally correlated with the corresponding block of the vectorial response variable. Given a multiway data set $\underline{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$

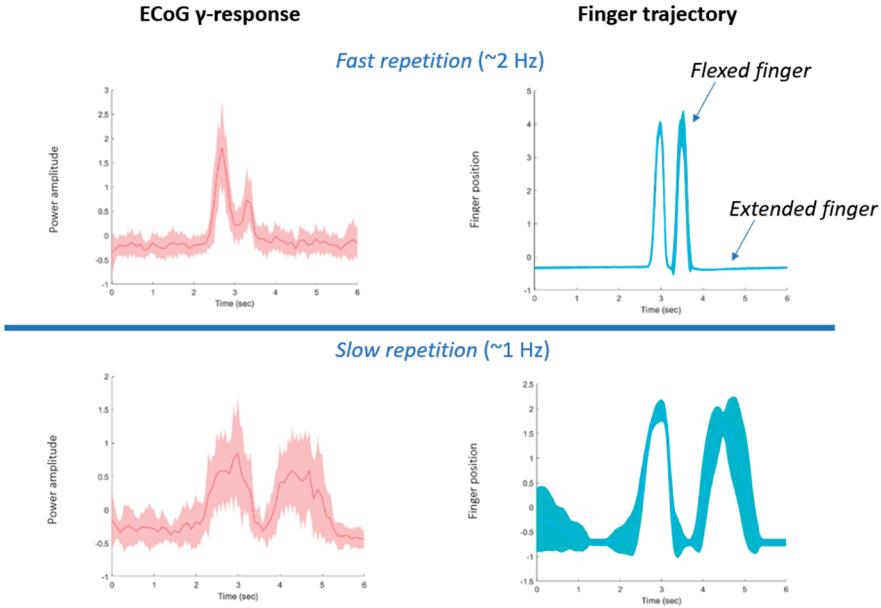


Figure 1.5: Evidence of nonlinear ECoG responses in high gamma band (γ_3 , see further). Top row: When performing fast index finger repetitions (~2Hz) (right panel), the second γ_3 peak is lower than the first one (left panel). Bottom row: When performing slow repetitions (~1Hz, right panel), the peaks are similar (left panel). Thick lines represent averages over 30 trials, shaded areas standard deviations. Spectral results for 5 Hz sampling rate. Data set from [52]

and vectoral response $\mathbf{y}_{\text{train}} \in \mathbb{R}^{I_1}$, BTTR training consists of automatically identifying K blocks s.t.:

$$\underline{\mathbf{X}}_{\text{train}} = \sum_{k=1}^K \underline{\mathbf{G}}_k \times_1 \mathbf{t}_k \times_2 \mathbf{P}_k^{(2)} \times_3 \dots \times_N \mathbf{P}_k^{(n)} + \underline{\mathbf{E}}_k$$

$$\mathbf{y}_{\text{train}} = \sum_{k=1}^K \mathbf{u}_k + \mathbf{f}_k \text{ with } \mathbf{u}_k = \mathbf{t}_k b_k$$

with $\underline{\mathbf{G}}_k \in \mathbb{R}^{1 \times R_2^k \times \dots \times R_N^k}$ the core tensor for the kth-block, $\mathbf{P}_k^{(n)}$ the kth loading matrix for the n-mode, \mathbf{u}_k and \mathbf{t}_k the latent vectors, b_k the regression coefficient, and $\underline{\mathbf{E}}_k$ and \mathbf{f}_k the residuals. Once the model is trained – and, hence, $\underline{\mathbf{G}}_k$,

$\mathbf{P}_k^{(n)}$ and b_k are computed – the final prediction is obtained as: $\mathbf{y}_{\text{test}} = \mathbf{T}\mathbf{b} = \mathbf{X}_{\text{test}(1)}\mathbf{W}\mathbf{b}$ where each column $\mathbf{w}_k = (\mathbf{P}_k^{(n)} \otimes \dots \otimes \mathbf{P}_k^{(2)})\text{vec}(\underline{\mathbf{G}}_k)$. In our study [53], BTTR was shown to outperform other linear multiway regression models as well as more traditional ones when decoding offline performed single finger movements from ECoG recordings in humans (up to 90% Pearson correlation). We also evaluated the time needed for training these models and observed that BTTR was faster to train than all other models. However, despite these encouraging results, BTTR exhibited some limitations. For instance, when testing BTTR on rapid finger repetitions, decoding performance degraded. We attribute this to the fact that the ECoG-trajectory relationship became nonlinear.

We also developed the extended Block-Term Regression (eBTTR) model which is based on (L_1^k, \dots, L_N^k) BTD with automatic MTR determination [54]. Specifically, it is a deflation-based method that sequentially decomposes $\underline{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and vectoral response $\mathbf{Y}_{\text{train}} \in \mathbb{R}^{I_1 \times M}$ into a series of blocks of maximally correlated representations extracted s.t.

$$\begin{aligned}\underline{\mathbf{X}}_{\text{train}} &= \sum_{k=1}^K \underline{\mathbf{G}}_k \times_1 \mathbf{t}_k \times_2 \mathbf{P}_k^{(2)} \times_3 \dots \times_N \mathbf{P}_k^{(n)} + \underline{\mathbf{E}}_k \\ \mathbf{Y}_{\text{train}} &= \sum_{k=1}^K \mathbf{u}_k \mathbf{q}_k^T + \mathbf{F}_k \text{ with } \mathbf{u}_k = \mathbf{t}_k b_k\end{aligned}$$

with $\underline{\mathbf{G}}_k \in \mathbb{R}^{1 \times R_2^k \times \dots \times R_N^k}$ the core tensor for the kth-block, $\mathbf{P}_k^{(n)}$ the kth loading matrix for the n-mode, \mathbf{u}_k and \mathbf{t}_k latent components, \mathbf{q}_k the loading matrix, b_k the regression coefficient, and $\underline{\mathbf{E}}_k$ and \mathbf{F}_k residuals. Once the model is trained –and, hence, $\underline{\mathbf{G}}_k$, $\mathbf{P}_k^{(n)}$ and b_k are computed– the final prediction is obtained as follows: $\mathbf{Y}_{\text{test}} = \mathbf{T}\mathbf{Z} = \mathbf{X}_{\text{test}(1)}\mathbf{W}\mathbf{Z}$ where each column $\mathbf{w}_k = (\mathbf{P}_k^{(n)} \otimes \dots \otimes \mathbf{P}_k^{(2)})\text{vec}(\underline{\mathbf{G}}_k)$ and each row $z_k = b_k q_k$.

1.6.2 organisation

The thesis is organized as follows:

- **Chapter 2** *Source space reduction for eLORETA* We introduce Sparse eLORETA, a novel method for estimating a nonparametric solution to the source localization problem in scalp EEG which we use to unveil the cortical network involved in a cued right hand movement task. The goal of Sparse eLORETA is to generate a sparser solution compared to other

source localization methods including eLORETA while benefitting from the latter's superior source localization accuracy. Sparse eLORETA starts by reducing the source space of the Lead Field Matrix using Structured Sparse Bayesian Learning (SSBL) from which a Reduced Lead Field Matrix is constructed, which is used as input to eLORETA. With Sparse eLORETA, source sparsity can be traded against signal fidelity; the proposed optimum is shown to yield a much sparser solution than eLORETA's with only a slight loss in signal fidelity. When pursuing a data-driven approach, for cases where it is difficult to choose specific regions of interest (ROIs), or when subsequently a connectivity analysis is performed, source space reduction could prove beneficial. This work is published as [55].

- **Chapter 3** *Neural networks for directed connectivity estimation in source-reconstructed EEG data* Directed connectivity between brain sources identified from scalp EEG can shed light on the brain's information flows in motor-, cognitive-, sensory- and other tasks and even be used as a biomarker of neurological disorders. However, as volume conductance results in scalp activity being a mix of activities originating from multiple sources, the correct interpretation of their connectivity is a formidable challenge despite source localization being applied with some success. Traditional approaches to connectivity estimation rely on statistical assumptions that usually do not hold for scalp EEG, calling for a model-free approach. We investigated several types of Artificial Neural Networks (ANNs) in estimating directed connectivity between reconstructed EEG sources and assessed their accuracy with respect to several ground truth cases. We show that a Long Short-Term Memory (LSTM) neural network with non-uniform embedding yields the most promising results due to its relative robustness to differing dipole locations. We conclude that certain network architectures can compete with the already established methods for brain connectivity analysis. This work is published as [56].
- **Chapter 4** *Single finger trajectory prediction from intracranial brain activity using Block-Term Tensor Regression with fast and automatic component extraction* Multiway-or tensor-based decoding techniques for BCIs are believed to better account for the multilinear structure of ECoG signals than conventional vector-or matrix-based ones. However, despite their outlook on significant performance gains, the used parameter optimization approach is often too computationally demanding so that conventional techniques are still preferred. We propose two novel tensor factorizations which we integrate into our block-term tensor regression (BTTR) algorithm and further introduce a marginalization procedure that guarantees robust predictions while reducing the risk of overfitting (generalized regression). BTTR accounts for the underlying (hidden)

data structure in a fully automatic and computationally efficient manner, leading to a significant performance gain over conventional vector-or matrix-based techniques. As a challenging real-world application, we apply BTTR to accurately predict single finger movement trajectories from ECoG recordings. We compare the obtained performance with that of the state-of-the-art. This work is published as [53].

- **Chapter 5** *Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression* We introduce extended Block-Term Tensor Regression (eBTTR), a novel regression method designed to account for the multiway nature of human finger movement ECoG recordings. The proposed method relies on recursive Tucker decomposition combined with automatic component extraction. eBTTR outperforms state-of-the-art regression approaches, including multilinear and deep learning ones, in accurately predicting finger trajectories as well as unintentional finger coactivations. eBTTR rivals state-of-the-art approaches while being less computationally expensive, which is an advantage when ECoG electrodes are implanted acutely, as part of the refractory epilepsy patient's presurgical workup, limiting the time for decoder development and testing. This work is published as [54].

Finally, in **Chapter 6**, *Conclusion*, we summarize and discuss the conclusions reached in the previous chapters and provide some remarks on future applications of multiway ECoG-based BCI.



Figure 1.6: Left Panel: Digital data glove. Right Panel: Hand exoskeleton.

Bibliography

- [1] *Who spinal cord injury*, <https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury>, Accessed: 2022-06-30.
- [2] R. E. d. O. S. Ascari, R. Pereira, and L. Silva, “Mobile interaction for augmentative and alternative communication: A systematic mapping”, *Journal on Interactive Systems*, vol. 9, no. 2, 2018.
- [3] V. Gilja, C. Pandarinath, C. H. Blabe, P. Nuyujukian, J. D. Simeral, A. A. Sarma, B. L. Sorice, J. A. Perge, B. Jarosiewicz, L. R. Hochberg, *et al.*, “Clinical translation of a high-performance neural prosthesis”, *Nature medicine*, vol. 21, no. 10, pp. 1142–1145, 2015.
- [4] M. J. Vansteensel, E. G. Pels, M. G. Bleichner, M. P. Branco, T. Denison, Z. V. Freudenburg, P. Gosselaar, S. Leinders, T. H. Ottens, M. A. Van Den Boom, *et al.*, “Fully implanted brain–computer interface in a locked-in patient with als”, *New England Journal of Medicine*, vol. 375, no. 21, pp. 2060–2066, 2016.
- [5] A. B. Ajiboye, F. R. Willett, D. R. Young, W. D. Memberg, B. A. Murphy, J. P. Miller, B. L. Walter, J. A. Sweet, H. A. Hoyen, M. W. Keith, *et al.*, “Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration”, *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, 2017.
- [6] C. E. Bouton, A. Shaikhouni, N. V. Annetta, M. A. Bockbrader, D. A. Friedenberg, D. M. Nielson, G. Sharma, P. B. Sederberg, B. C. Glenn, W. J. Mysiw, *et al.*, “Restoring cortical control of functional movement in a human with quadriplegia”, *Nature*, vol. 533, no. 7602, pp. 247–250, 2016.
- [7] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia”, *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.

- [8] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, “High-performance neuroprosthetic control by an individual with tetraplegia”, *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.
- [9] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, “Direct cortical control of 3d neuroprosthetic devices”, *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.
- [10] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt, *et al.*, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm”, *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.
- [11] B. Wodlinger, J. Downey, E. Tyler-Kabara, A. Schwartz, M. Boninger, and J. Collinger, “Ten-dimensional anthropomorphic arm control in a human brain-machine interface: Difficulties, solutions, and limitations”, *Journal of neural engineering*, vol. 12, no. 1, p. 016011, 2014.
- [12] A. L. Benabid, T. Costecalde, A. Eliseyev, G. Charvet, A. Verney, S. Karakas, M. Foerster, A. Lambert, B. Morinière, N. Abroug, *et al.*, “An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: A proof-of-concept demonstration”, *The Lancet Neurology*, vol. 18, no. 12, pp. 1112–1122, 2019.
- [13] G. Schalk, “Can electrocorticography (ecog) support robust and powerful brain-computer interfaces?”, *Frontiers in neuroengineering*, p. 9, 2010.
- [14] W. Wang, J. L. Collinger, A. D. Degenhart, E. C. Tyler-Kabara, A. B. Schwartz, D. W. Moran, D. J. Weber, B. Wodlinger, R. K. Vinjamuri, R. C. Ashmore, *et al.*, “An electrocorticographic brain interface in an individual with tetraplegia”, *PloS one*, vol. 8, no. 2, e55344, 2013.
- [15] E. C. Leuthardt, G. Schalk, J. R. Wolpaw, J. G. Ojemann, and D. W. Moran, “A brain-computer interface using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 1, no. 2, p. 63, 2004.
- [16] N. R. Anderson, T. Blakely, G. Schalk, E. C. Leuthardt, and D. W. Moran, “Electrocorticographic (ecog) correlates of human arm movements”, *Experimental brain research*, vol. 223, no. 1, pp. 1–10, 2012.
- [17] T. Ball, M. Kern, I. Mutschler, A. Aertsen, and A. Schulze-Bonhage, “Signal quality of simultaneously recorded invasive and non-invasive eeg”, *Neuroimage*, vol. 46, no. 3, pp. 708–716, 2009.
- [18] D. T. Bundy, M. Pahwa, N. Szrama, and E. C. Leuthardt, “Decoding three-dimensional reaching movements using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 13, no. 2, p. 026021, 2016.
- [19] B. Horwitz, “The elusive concept of brain connectivity”, *Neuroimage*, vol. 19, no. 2, pp. 466–470, 2003.

- [20] H. S. Lee, L. Schreiner, S.-H. Jo, S. Sieghartsleitner, M. Jordan, H. Pretl, C. Guger, and H.-S. Park, “Individual finger movement decoding using a novel ultra-high-density electroencephalography-based brain-computer interface system”, *Frontiers in Neuroscience*, vol. 16, 2022.
- [21] W. Penny. “Meg and eeg analysis”. (2018), [Online]. Available: <https://www.fil.ion.ucl.ac.uk/~wpenny/research/meeg.html> (visited on 08/04/2018).
- [22] B. J. Roth, M. Balish, A. Gorbach, and S. Sato, “How well does a three-sphere model predict positions of dipoles in a realistically shaped head?”, *Electroencephalography and clinical Neurophysiology*, vol. 87, no. 4, pp. 175–184, 1993.
- [23] B. N. Cuffin, “A method for localizing eeg sources in realistic head models”, *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 1, pp. 68–71, 1995.
- [24] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT press, 2014.
- [25] K. Hoechstetter, H. Bornfleth, D. Weckesser, N. Ille, P. Berg, and M. Scherg, “Besa source coherence: A new method to study cortical oscillatory coupling”, *Brain topography*, vol. 16, no. 4, pp. 233–238, 2004.
- [26] A. Delorme and S. Makeig, “Eeglab: An open source toolbox for analysis of single-trial eeg dynamics including independent component analysis”, *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004.
- [27] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, “Fieldtrip: Open source software for advanced analysis of meg, eeg, and invasive electrophysiological data”, *Computational intelligence and neuroscience*, vol. 2011, p. 1, 2011.
- [28] R. D. Pascual-Marqui *et al.*, “Standardized low-resolution brain electromagnetic tomography (sloreta): Technical details”,
- [29] J. Groß, J. Kujala, M. Hämäläinen, L. Timmermann, A. Schnitzler, and R. Salmelin, “Dynamic imaging of coherent sources: Studying neural interactions in the human brain”, *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 694–699, 2001.
- [30] B. D. Van Veen, W. Van Drongelen, M. Yuchtman, and A. Suzuki, “Localization of brain electrical activity via linearly constrained minimum variance spatial filtering”, *IEEE Transactions on biomedical engineering*, vol. 44, no. 9, pp. 867–880, 1997.
- [31] B. Van Veen and K. Buckley, “Beamforming: A versatile approach to spatial filtering”, *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988. DOI: [10.1109/53.665](https://doi.org/10.1109/53.665).
- [32] M. F. Hnazaee, M. Verwoert, Z. V. Freudenburg, S. M. van der Salm, E. J. Aarnoutse, S. Leinders, M. M. Van Hulle, N. F. Ramsey, and M. J. Vansteensel, “Towards predicting ecog-bci performance: Assessing

- the potential of scalp-eeg”, *Journal of Neural Engineering*, vol. 19, no. 4, p. 046 045, 2022.
- [33] R. Grech, T. Cassar, J. Muscat, K. P. Camilleri, S. G. Fabri, M. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vanrumste, “Review on solving the inverse problem in eeg source analysis”, *Journal of neuroengineering and rehabilitation*, vol. 5, no. 1, pp. 1–33, 2008.
 - [34] M. Van Vliet, N. Chumerin, S. De Deyne, J. R. Wiersema, W. Fias, G. Storms, and M. M. Van Hulle, “Single-trial erp component analysis using a spatiotemporal lcmv beamformer”, *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 1, pp. 55–66, 2015.
 - [35] B. Wittevrongel and M. M. Van Hulle, “Faster p300 classifier training using spatiotemporal beamforming”, *International journal of neural systems*, vol. 26, no. 03, p. 1650 014, 2016.
 - [36] ——, “Spatiotemporal beamforming: A transparent and unified decoding approach to synchronous visual brain-computer interfacing”, *Frontiers in neuroscience*, vol. 11, p. 630, 2017.
 - [37] A. Van Den Kerchove, A. Libert, B. Wittevrongel, and M. M. Van Hulle, “Classification of event-related potentials with regularized spatiotemporal lcmv beamforming”, *Applied Sciences*, vol. 12, no. 6, p. 2918, 2022.
 - [38] A. Libert, A. Van Den Kerchove, B. Wittevrongel, and M. M. Van Hulle, “Analytic beamformer transformation for transfer learning in motion-onset visual evoked potential decoding”, *Journal of Neural Engineering*, vol. 19, no. 2, p. 026 040, 2022.
 - [39] A. Libert, B. Wittevrongel, F. Camarrone, and M. M. Van Hulle, “Phase-spatial beamforming renders a visual brain computer interface capable of exploiting eeg electrode phase shifts in motion-onset target responses”, *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 5, pp. 1802–1812, 2021.
 - [40] J. Kubanek, K. J. Miller, J. G. Ojemann, J. R. Wolpaw, and G. Schalk, “Decoding flexion of individual fingers using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 6, no. 6, p. 066 001, 2009.
 - [41] Z. Wang, Q. Ji, K. J. Miller, and G. Schalk, “Decoding finger flexion from electrocorticographic signals using a sparse gaussian process”, in *2010 20th International conference on pattern recognition*, IEEE, 2010, pp. 3756–3759.
 - [42] R. Flamary and A. Rakotomamonjy, “Decoding finger movements from ecog signals using switching linear models”, *Frontiers in neuroscience*, vol. 6, p. 29, 2012.
 - [43] N. Liang and L. Bougrain, “Decoding finger flexion from band-specific ecog signals in humans”, *Frontiers in neuroscience*, vol. 6, p. 91, 2012.

- [44] Z. Xie, O. Schwartz, and A. Prasad, “Decoding of finger trajectory from ecog using deep learning”, *Journal of neural engineering*, vol. 15, no. 3, p. 036009, 2018.
- [45] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis”, *IEEE signal processing magazine*, vol. 32, no. 2, pp. 145–163, 2015.
- [46] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications”, *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [47] L. De Lathauwer and D. Nion, “Decompositions of a higher-order tensor in block terms—part iii: Alternating least squares algorithms”, *SIAM journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.
- [48] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, “Higher order partial least squares (hopls): A generalized multilinear regression method”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1660–1673, 2012.
- [49] A. Eliseyev and T. Aksanova, “Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording”, *PloS one*, vol. 11, no. 5, e0154878, 2016.
- [50] Z. C. Chao, Y. Nagasaka, and N. Fujii, “Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey”, *Frontiers in neuroengineering*, vol. 3, p. 3, 2010.
- [51] M. P. Branco, Z. V. Freudenburg, E. J. Aarnoutse, M. G. Bleichner, M. J. Vansteensel, and N. F. Ramsey, “Decoding hand gestures from primary somatosensory cortex using high-density ecog”, *NeuroImage*, vol. 147, pp. 130–142, 2017.
- [52] J. C. Siero, D. Hermes, H. Hoogduin, P. R. Luijten, N. F. Ramsey, and N. Petridou, “Bold matches neuronal activity at the mm scale: A combined 7 t fmri and ecog study in human sensorimotor cortex”, *Neuroimage*, vol. 101, pp. 177–184, 2014.
- [53] A. Faes, F. Camarrone, and M. M. Van Hulle, “Single finger trajectory prediction from intracranial brain activity using block-term tensor regression with fast and automatic component extraction”, *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [54] A. Faes and M. Van Hulle, “Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression”, *Journal of Neural Engineering*, 2022.
- [55] A. Faes, A. de Borman, and M. Van Hulle, “Source space reduction for eloreta”, *Journal of Neural Engineering*, vol. 18, no. 6, p. 066014, 2021.

- [56] A. Faes, I. Vantieghem, and M. M. Van Hulle, “Neural networks for directed connectivity estimation in source-reconstructed eeg data”, *Applied Sciences*, vol. 12, no. 6, p. 2889, 2022.

Chapter 2

Source Space Reduction for eLORETA

A. FAES, A. DE BORMAN AND M.M. VAN HULLE

Journal of Neural Engineering 18.6 (2021): 066014

2.1 Abstract

Objective We introduce Sparse eLORETA, a novel method for estimating a nonparametric solution to the source localization problem. Its goal is to generate a sparser solution compared to other source localization methods including eLORETA while benefitting from the latter's superior source localization accuracy.

Approach Sparse eLORETA starts by reducing the source space of the Lead Field Matrix using Structured Sparse Bayesian Learning (SSBL) from which a Reduced Lead Field Matrix is constructed, which is used as input to eLORETA.

Main results With Sparse eLORETA, source sparsity can be traded against signal fidelity; the proposed optimum is shown to yield a much sparser solution than eLORETA's with only a slight loss in signal fidelity.

Significance When pursuing a data-driven approach, for cases where it is difficult to choose specific regions of interest (ROIs), or when subsequently a connectivity analysis is performed, source space reduction could prove beneficial.

2.2 Introduction

It is commonly accepted that perception, motor behavior, language, and cognition emerge from coordinated information flows between functionally specialized regions. As these activities and flows are dynamic, imaging techniques such as MEG and (scalp-based) EEG are preferred over spatially more accurate yet temporally less detailed ones such as fMRI and PET [1], [2].

When localizing active brain regions, MEG and EEG signals need to be projected back into the brain, a procedure called source localization [3], [4]. It requires solving an inverse problem (from scalp electrode- or MEG sensor space to 3D brain space) that is bound by the recordings not providing sufficient information to guarantee a unique and stable solution [5]. The recordings reflect mixtures of local neural generators of which the activity is spatially smeared out due to volume conduction, causing localization errors or even a failure of the localization procedure.

In the case of EEG, source localization starts with a forward equation expressing the scalp electric potential differences as a function of the current densities produced by the neural generators, an easy to solve problem using linear algebra. The difficulty arises when the scalp potential differences are known and we need to solve the inverse solution for unknown amplitudes and orientations of the current densities [6]. Because there are many more unknowns than equations, additional assumptions are needed, leading to several algorithmic approaches. Two mainstream source localization techniques have been proposed: the equivalent current dipole (parametric) and the distributed source models (nonparametric) [6]. The latter differs from the former in that it does not require the prior specification of the number of neural sources. The former solves the inverse solution to obtain the location and orientation of these sources. The latter assumes the entire brain volume to consist of fixed source locations and solves the inverse solution to estimate their amplitudes; as these sources are by no means assumed to be independent, they cannot be regarded as different neural generators. There is a tendency to rely on distributed source modeling when there is no prior information about the expected active brain regions, in the opposite case about the equivalent current dipoles. Within the realm of the nonparametric approaches, several source localization methods have been proposed.

Linearly Constrained Minimum Variance (LCMV) [7], [8] is a spatial filtering method that accepts activity originating from a specific location, while attenuating that from other locations, thus a way of beamforming. The result is an estimate of the activity generated by the target brain locations. LCMV assumes that the sources are not temporally correlated. The inverse filter is based on minimizing the source variance at a given location, subject to a ‘unit-gain constraint’. The mapping between multiple sources and scalp-EEG is obtained by combining corresponding LCMV filters.

Exact Low Resolution Electromagnetic Tomography (eLORETA) [9] is a linear inverse method characterized by a spatially smooth current density. It belongs to the LORETA family with zero localization error in the case of one active dipole and no noise. Several studies in realistic settings (multiple sources, noise, actual recordings) reported that eLORETA outperforms other linear methods in localization accuracy [10].

Brain Connectivity Variable Resolution Tomographic Analysis (BC-VARETA) is a recent inverse method developed by Gonzales et al. [11] that has already been adopted in a few studies [12]–[14]. The method estimates the inverse solution and its precision matrix, which represents the connectivity parameters, by using the frequency domain representation of the stationary time series. It belongs to the category of parametric source localization methods. As BC-VARETA is computationally expensive, to reduce source space dimensionality, a prior screening of the sources is performed using Structured Sparse Bayesian Learning (SSBL), which has a parameter that controls to what extent (in %) source space is reduced, called the sparsity constraint (κ). If it is set to 5%, then source space is reduced to 5% of its original size [15]. The extent of dimensionality reduction is controlled by an upper bound on the number of selected sources so that κ is chosen depending on the context. One should note that BC-VARETA localizes sources but does not reconstruct their time series. However, it has several interesting features, one of which is the much sparser solution it produces than mainstream source localization techniques. When applied to the Cuban Brain Mapping Project, BC-VARETA’s results have been shown to be in alignment with the physiology of human resting state EEG in the spectral domain [11].

In this paper, we propose a novel method for source localization from multiple stationary time series. The method aims at a sparser solution compared to other source localization methodologies while maintaining localization accuracy. We first introduce our source localization method called Sparse eLORETA. Then, in Section 2.4, we apply the proposed method to one simulated- and 2 real-world EEG datasets, the Cuban Human Brain Project, and the Berlin Brain Connectivity Benchmark and Data Analysis Challenge Dataset, and report the

results. Finally, we summarize and discuss our results and draw our main conclusion.

2.3 Materials and Methods

When pursuing a data-driven approach, for cases where it is difficult to choose a priori regions of interest (ROIs), or when subsequently a connectivity analysis is performed, source space reduction could prove beneficial. The advantage of BC-VARETA in this respect is that it generates a sparse, yet accurate source localization. This motivated us to utilize several components from BC-VARETA to sparsify source localization. Conceptually, we use the source localization map obtained from BC-VARETA as a mask for eLORETA. In what follows we briefly discuss BC-VARETA. Implementation details, as well as of the other used methods, are listed in the Supplementary Materials section.

First, we apply SSBL to reduce the source space of the Lead Field Matrix. SSBL extracts the possibly active generators and gives us the indices of dipoles that are potentially active. These indices are used to reconstruct a new, partial Lead Field Matrix. Finally, the resulting Lead Field Matrix is used within eLORETA to generate the inverse solution.

Algorithm 1: Source Space Reduction for eLORETA.

```
[miu] ← ssbl(Svv, HM, seg, dip_map);
[indms] ← smoothing(miu, κ * dip, vert, fac, ind);
ReducedHM ← HM(:, indms);
P ← eLORETA(ReducedHM, γ);
```

In Algorithm 1, we list the pseudocode behind Sparse eLORETA. First, we apply SSBL to reduce the source space of the Lead Field Matrix. SSBL requires the Covariance Matrix of the input data (Svv), the head model (HM), the number of epochs (seg) and a dictionary mapping each dipole onto itself (i.e. $1 \rightarrow 1$). SSBL extracts the possibly active generators as a list of indices (miu). In the next step, a smoothing effect is applied to the results from SSBL. Here we use the κ parameter to indicate to what percentage of all dipoles (dip indicates the number of dipoles) smoothing should prune dipoles. When taken too small, relevant dipoles could be pruned, when taken too large irrelevant dipoles could remain. Note that the smoothing function also requires the location of all dipoles ($vertices$), the triangulation (fac), and the range of all dipole indices (ind). The smoothing step returns the indices of the remaining, potentially active dipoles, and this concludes source space reduction. These indices are used to construct the reduced head model that is input to eLORETA, which

generates the inverse solution P . eLORETA takes only one other parameter, γ , which is the regularization parameter.

2.3.1 Simulated EEG Dataset

A simulation study provides us with a ground truth against which our results can be verified and compared with those of other methods. We adopt the simulation framework of Anzolin et al. which consists of several steps: the generation of brain signals with a predetermined connectivity pattern as ground truth, noise generation, forward modeling, inverse modeling, connectivity estimation, and performance evaluation [16]. However, we propose a different approach for the first step, as we are not interested in connectivity estimation.

Pseudo-EEG data were generated by employing the toolbox described in the work from Haufe et al. [17]. We simulated three time series of 500 samples. A 3-dimensional multivariate autoregressive (MVAR) model of order 2 was used to simulate pseudo-EEG data. This process has previously been used to evaluate non-stationary, directed interactions in multivariate neural data [18]. Hereby, x_1 , x_2 and x_3 represent the electrical activity of three dipoles within the brain. Each one is an active source contributing to the pseudo-EEG measured on the scalp assuming an EEG cap with 108 electrodes.

$$\begin{cases} x_1(n) = 0.5x_1(n-1) - 0.7x_1(n-2) \\ \quad + 0.25x_2(n-1) + w_1(n) \\ x_2(n) = 0.7x_2(n-1) - 0.5x_2(n-2) \\ \quad + 0.2x_1(n-1) + 0.25x_3(n-1) + w_2(n) \\ x_3(n) = 0.8x_3(n-1) + w_3(n) \end{cases} \quad (2.1)$$

In order to probe the robustness of source localization, we considered not only brain sources but also 500 noise sources as background activity. We considered multiple types of noise: (incoherent) pink noise (i.e., standard pink noise), coherent autoregressive noise [19], and coherent pink noise [20], and gauged their effect on source localization accuracy.

BC-VARETA requires information about the cortex mesh. We have used the New York Head Model available in the Berlin Brain Connectivity Benchmark and Data Analysis Challenge with mesh information in terms of vertex coordinates and triangles or faces [21]. We introduced 2004 electric equivalent dipoles, homogeneously distributed and located using the ICBM152v2009 stereotaxic registration model. The 2004 dipole positions were obtained by subsampling the 75000 MNI coordinates available in the New York Head Model [22].

For each simulation, a different triple dipole configuration was created by having two dipoles at fixed positions and the third dipole at a randomly chosen one. The

Close Superficial	(33.92, 59.40, -3.27)
Far Superficial	(25.01, -93.32, 7.55)
Close Deep	(-2.78, -2.56, 4.47)
Far Deep	(58.19, -32.33, 33.18)
Close Superficial	(40.87, 59.98, 5.69)
Far Superficial	(40.87, 59.98, 5.69)
Close Deep	(-5.88, -34.51, 34.22)
Far Deep	(-30.73, 2.01, -6.17)

Table 2.1: Fixed dipole coordinates within the subsampled New York head model, for each configuration. The first table lists the positions of the first fixed dipole, the second table those of the second fixed dipole.

fixed dipoles were positioned in four configurations using the same stereotactical coordinates as in [16]: two superficial dipoles (distance from the origin > 6.5 cm), two deep ones (distance from the origin < 6 cm), or two close together (relative distance < 5 cm) or far apart (relative distance > 8 cm) (Table 2.1). We further assumed that the dipoles are oriented perpendicularly to the cortical surface. The 500 noise sources were randomly distributed within the brain mesh excluding the three active dipole locations.

Once we activate the 3 brain sources and the 500 noise sources, we can project their activities to the 108 EEG electrodes. Once projected, the scalp activity of the brain source $x^{active}(t)$ and the scalp activity of the noise sources $x^{noise}(t)$ are summed. This is done with a coefficient λ , set to 0.5. $\|x^{active}(t)\|_F$ and $\|x^{noise}(t)\|_F$ are the Frobenius norms of the multivariate time series $x^{active}(t)$ and $x^{noise}(t)$, respectively, i.e., the square root of the squared activity summed spatially and temporally. The Frobenius norm is used to scale the brain source and noise activities:

$$x^{brain}(t) = \lambda \times \frac{x^{active}(t)}{\|x^{active}(t)\|_F} + (1 - \lambda) \times \frac{x^{noise}(t)}{\|x^{noise}(t)\|_F}. \quad (2.2)$$

The signal definition, seen in equation 2.2, implies that deep active dipoles can have the same strength as shallow ones. This could make the discovery of deep active dipoles more difficult but, on the other hand, as the Signal to Noise ratio (SNR) is computed not per source, but for all three dipoles simultaneously, this imbalance is mitigated. Finally, in order to simulate the measurement noise, spatially and temporally uncorrelated signals, Gaussian-distributed uncorrelated white noise x^{m_noise} is added to $x^{brain}(t)$ with a predefined Mean Squared Amplitude ratio of 0.9. The overall pseudo-EEG data is defined by the following

equation:

$$x^{measured}(t) = 0.9 \times \frac{x^{brain}(t)}{\|x^{brain}(t)\|_F} + 0.1 \times \frac{x^{m_noise}(t)}{\|x^{m_noise}(t)\|_F}. \quad (2.3)$$

2.3.2 Real-world EEG Datasets

Two EEG datasets are considered as real-world cases. The first set was recorded from a 32 year old healthy male under resting state (eyes closed) condition using 128 channels of a MEDICID 5 system operating at 200 Hz sampling rate. It is part of the Cuban Brain Mapping Project [23], created in 2005 with the aim to obtain brain atlases of the Cuban population.

The second dataset consists of 118 channel EEG activity recorded from five healthy subjects [24] during a visually-cued multi-class motor imagery task (imagined left hand, right hand and right foot movements) at 100 Hz sampling rate, available under Dataset IVa of the BCI Competition III (http://www.bbci.de/competition/iii/desc_IVa.html). We consider the case of imagined right hand movement of subject ay. Prior to source localization, we apply a 4th order zero-phase Butterworth filter to extract alpha band activity (8-12 Hz) and cut out epochs from 0.5s after the start cue until 0.5s before stop cue to avoid contamination by the visual cue and early stopping, respectively. We determine within each epoch the degree of event-related desynchronization (ERD) or -synchronization (ERS) with respect to baseline activity. Baseline activity epochs are taken from the rest periods between trials starting 0.3s after the onset of a rest period until 0.3s before it ends. In total, there are 18 epochs (18 epochs for baseline activity and 18 epochs for imagined movement activity). These epochs are then concatenated.

Similar to performed movement, we expect imagined movement activity in motor areas including the primary motor cortex (M1) and supplementary motor area (SMA), and ventral and dorsal parts of the premotor cortex (PMC), although activation of M1 during imagined movement is weaker compared to that during performed movement [25]. Activity is expected in areas that relate to action planning, such as the dorsolateral prefrontal cortex (DLPFC), ventrolateral prefrontal cortex (VLPFC) (which anatomically corresponds to the inferior frontal cortex, IFC), and posterior parietal cortex (PPC) including the angular gyrus [26]. However, given the rather simple motor task, we expect less involvement of the medial prefrontal cortex (MPFC), such as SMA. As far as we know, no source localization results have been published for this dataset.

For both datasets, we have used the Head Model of the Cuban Brain Mapping Project [23]. It contains mesh information in terms of vertex coordinates and triangles or faces. We introduced 6003 homogeneously distributed electric equivalent dipoles.

	LCMV	eLORETA	Sparse eLORETA
PN	0.9867	0.9537	0.9234
CPN	0.9119	0.9329	0.9241
CAN	0.9807	0.9479	0.9020

Table 2.2: Average correlation coefficients of the Far Superficial dipole configuration given 3 noise models. PN = (incoherent) pink noise, CPN = coherent pink noise, CAN = autoregressive pink noise.

	LCMV	eLORETA	Sparse eLORETA
FS	0.9867	0.9537	0.9234
CS	0.9707	0.9750	0.9023
FD	0.9204	0.8833	0.7721
CD	0.8383	0.6297	0.5991

Table 2.3: Average correlation coefficients in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.

2.4 Results

2.4.1 Simulated EEG Dataset Results

The accuracy with which the active source signals are estimated with LCMV, eLORETA and Sparse eLORETA, for different source configurations and noise types, are summarized in Tables 2.2- 2.3. The tables list the average correlation coefficients between the reconstructed dipoles and ground truth signals (obtained from Eq 2.1), averaged over the 1000 randomly chosen positions of the third source with the other 2 sources kept fixed.

A first observation is that LCMV yields the best correlation coefficients, which is counter-intuitive since in real-world cases we expect eLORETA to outperform LCMV. eLORETA does not provide an exact localization due to the presence of noise models in the simulation. There is a slight decrease in correlation obtained with Sparse eLORETA compared to eLORETA. This is a side effect of our source space reduction procedure: we found that, by increasing the sparsity constraint κ , the correlation coefficient increases and from $\kappa = 0.12$ onwards coincides with eLORETA's (blue line in Figure 2.1). However, κ also affects source localization accuracy as reported next.

Figure 2.2 shows the source distributions obtained with eLORETA, BC-VARETA and Sparse eLORETA for (uncoherent) pink noise in the case of the Far Superficial dipole configuration. We observe that, despite LCMV's higher

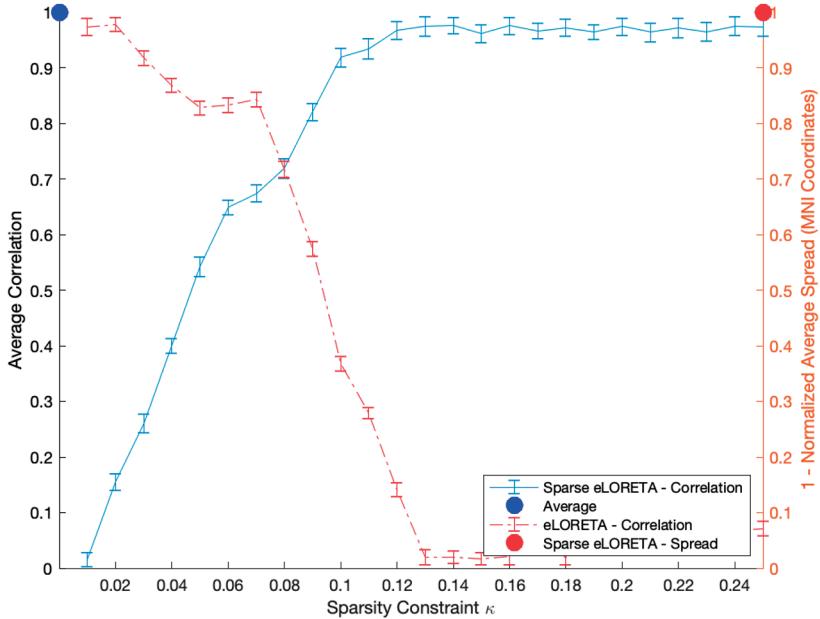


Figure 2.1: Average correlation (blue line, left scale) and complement of the average normalized spatial spread (red line, right scale), normalized with respect to eLORETA’s outcome (red ball), plotted as a function of the sparsity constraint κ . The blue ball denotes eLORETA’s average correlation. The Pareto-optimal solution is at the intersection ($\kappa \approx 0.08$).

signal correlation (Table 2.3), the reconstructed dipoles are dispersed without showing any evidence of clustering around the true ones. With eLORETA, the found dipoles are clustered around the true ones but still less than with BC-VARETA. Compared to eLORETA, we clearly see the effect of source space reduction: the reconstructed dipoles are grouped and close to the true ones. Results are analogous for other configurations given the quantitative results listed in Table 2.3.

We also assessed the computational effort (i.e. elapsed time) required by the source localization methods (Table 2.4). We observe that BC-VARETA takes much more time compared to the other methods. Although the SSBL step is time-consuming, Sparse eLORETA requires only slightly more time than eLORETA, a beneficial side effect of reducing source space.

	Timing
LCMV	0.0286
eLORETA	0.3567
Sparse eLORETA	1.0459
BC-VARETA	68.0538

Table 2.4: Average elapsed time (in seconds) of LCMV, eLORETA, Sparse eLORETA and BC-VARETA, determined using Matlab's tic toc function, averaged over 1000 runs.

2.4.2 Clustering reconstructed dipoles

The results reveal major differences in the spatial distributions of reconstructed sources returned by the different methods. In order to quantify these distributions, we applied medoid clustering [27] to partition the reconstructed sources into clusters, extract their centers ("medoids") and determine their "spatial spread" defined as the average distance of the cluster members to their corresponding medoids. Note that medoid clustering assumes the number of clusters is known. We have applied the method to cluster the eLORETA, Sparse eLORETA, BC-VARETA and LCMV reconstructed source distributions. Figure 2.2 shows the result of the Far Superficial configuration in the presence of (uncoherent) pink noise. The Supplementary Materials Section shows the implementation details for the mediod clustering.

Table 2.5 lists, for all dipole configurations considered, the average spatial spread, averaged over the 3 clusters. The first point to note is the widespread nature of the LCMV dipole solutions. We also observe that Sparse eLORETA consistently yields smaller cluster sizes compared to eLORETA yet slightly larger than BC-VARETA's. Table 2.6 shows the average distance between the cluster elements and the true sources. We observe that the performances of eLORETA, Sparse eLORETA and BC-VARETA are very similar. In line with the (average) spatial spread, LCMV tends to generate larger distances between medoids and true sources.

When plotting the complement of the spatial spread against the sparsity constraint κ (red line in Figure 2.1), we see that the larger κ , the smaller the complement of the spatial spread, or the larger the spatial spread, until about $\kappa = 0.12$ after which it levels off. When comparing this with the average correlation (blue line), we observe that $\kappa = 0.12$ is also the point where it levels off and, in addition, where there is no significant difference between eLORETA (blue ball) and sparse eLORETA. This is likely due to the fact that, for higher κ 's, the sparsity constraint only prunes irrelevant dipoles. Hence, there seems to be a tradeoff between signal correlation and spatial spread or, in other words,

	LCMV	eLORETA	Sparse eLORETA	BC-VA RETA
FS	1196.7	199.47	113.16	113.89
CS	1165.6	335.71	57.61	4.41
FD	4473.1	9239.1	175.34	4.48
CD	3001.4	907.3	242	65.4

Table 2.5: Average spatial spread in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.

	LCMV	eLORETA	Sparse eLORETA	BC-VA RETA
FS	68.71	11.88	11.64	11.43
CS	60.14	15.23	12.76	8.61
FD	19.24	41.46	37.07	35.28
CD	45.95	42.93	42.93	45.39

Table 2.6: Average distance between medoids and true sources in the presence of (uncoherent) pink noise for different dipole configurations. FS = Far Superficial, CS = Close Superficial, FD = Far Deep, CD = Close Deep.

between temporal and spatial accuracy. When normalizing the two metrics, thus, time series correlation and normalized complement of the average spread, they can be plotted on the same [0,1] scale. We take κ at the intersection as in this way we cannot improve on one metric without worsening on the other (Pareto optimum).

2.4.3 Real-world EEG Dataset Results

Figure 2.3 shows 300 source localizations (aka bootstrapping) for the first real world case (Cuban EEG dataset) as obtained with eLORETA, Sparse eLORETA, BC-VARETA and LCMV. We observe that the sources reconstructed with LCMV are more spread out. The most active sources (darker color) in Sparse eLORETA are in areas where BC-VARETA and Sparse eLORETA also locate their sources. Compared to eLORETA, we clearly observe the effect of source space reduction in Sparse eLORETA. There are no notable differences between Sparse eLORETA and BC-VARETA's results. For κ we took 0.08, as in the simulation setup, also given that it does not seem to prune sources from relevant regions while maintaining the desired sparsity.

Figure 2.4 shows 300 source localizations for the second real world set (subject ay, imagined right hand movement) obtained with eLORETA, Sparse eLORETA,

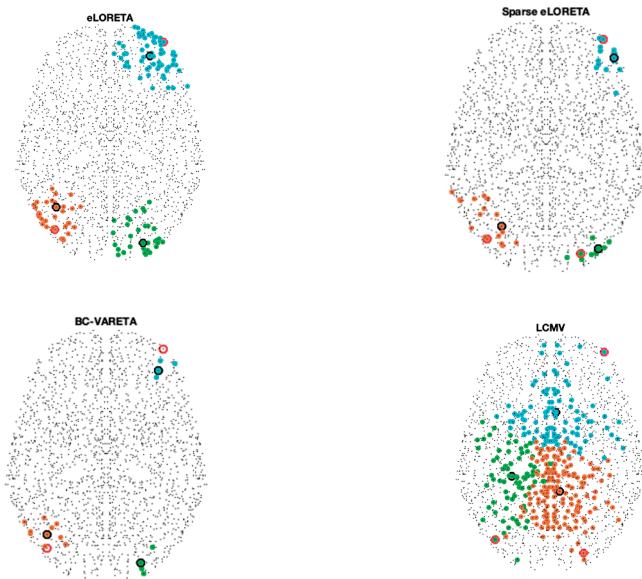


Figure 2.2: Distribution of reconstructed dipoles (brain top views, nose pointing upwards) of the simulation study, (uncoherent) pink noise case. Shown are the results obtained using eLORETA, Sparse eLORETA, BC-VARETA and LCMV given the Far Superficial configuration, with the 3 medoid clusters marked in orange, green and cyan. Medoids centers are indicated with black filled circles, the ground truth dipoles with red filled circles. In the bottom right cluster (green for eLORETA), the medoid center overlaps with the ground truth dipole.

BC-VARETA. We observe that BC-VARETA does not perform well on this dataset as only a few sources are localized. LCMV also has issues as practically all sources are labeled as ERS and distributed uniformly without showing any identifiable clusters (not shown). eLORETA, and Sparse eLORETA in particular, are able to reconstruct sources in the motor cortex. For the latter two methods, we observe alpha band ERS activity (i.e. deactivation) in DLFPC (bilaterally), angular gyrus (ANG, contralaterally), and superior PPC (contralateral in Sparse eLORETA, bilateral in eLORETA), and ERD activity in M1 (contralateral in Sparse eLORETA, bilateral in eLORETA albeit less ipsilateral) and VLPFC (bilaterally), but no activity in MPFC (SMA) (see Figure 2.5 for a labeling of the ERD/ERS active brain regions for Sparse eLORETA). Time-locked deactivations during goal-directed behavior in these regions have been observed in several studies and are considered part of a default mode network [28]. We observe that eLORETA exhibits close-to-zero ERD/ERS activities near the

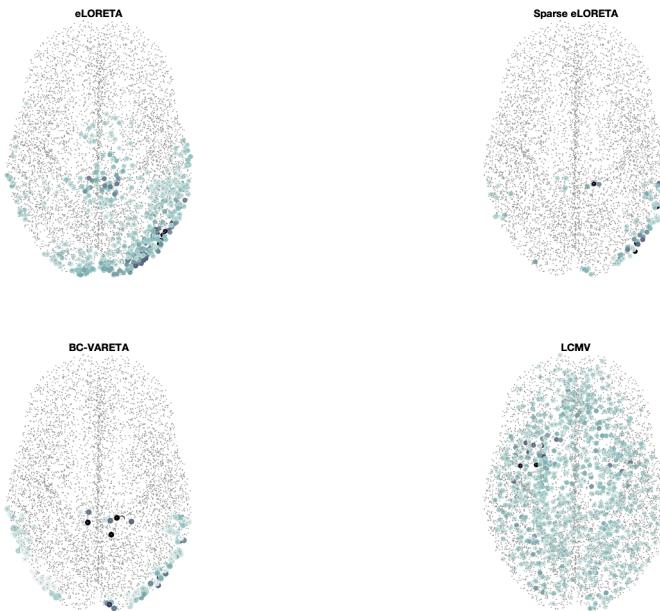


Figure 2.3: Distribution of reconstructed dipoles (brain top views, nose pointing upwards) of the Cuban EEG dataset using eLORETA, Sparse eLORETA, BC-VARETA and LCMV. The grey nodes in the background are mesh nodes. The filled circles are active dipoles with a darker color indicating more activity.

temporal pole and ERS activities along the inferior temporal gyrus bilaterally but all these have been pruned by SSBL in Sparse eLORETA.

2.5 Discussion

Localizing active brain regions from non-invasive recordings is a challenging problem as there is simply not enough information to arrive at a unique, stable solution without relying on additional assumptions. This has led to a gamut of localization methods at least in the case of scalp EEG, most importantly LCMV and several members of the LORETA family, such as eLORETA and sLORETA. Some methods in addition provide estimates of the functional connectivity between reconstructed sources as in BC-VARETA [11], where sources and connectivities are computed simultaneously. A more common approach is to first anatomically define well-separated ROIs and then to estimate their functional connectivity [29] with recent studies pushing for adaptive cortical

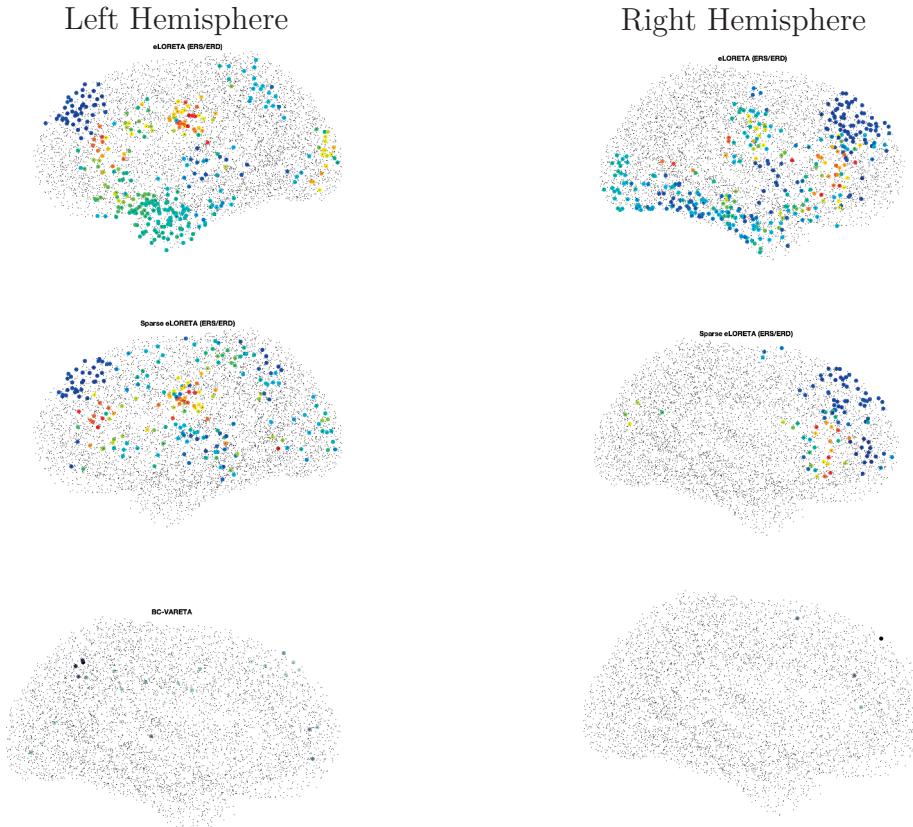


Figure 2.4: ERD/ERS post-cue imagined right hand movement activation of 300 sources (aka bootstrapping) obtained with eLORETA, Sparse eLORETA and BC-VARETA, mapped on left and right lateral brain views (arranged column-wise). The grey nodes in the background are mesh nodes. Blue dots correspond to ERS, yellow/red to ERD and green to neither. For BC-VARETA dipoles are grey dots as no ERS/ERS assessment is possible.

meshes to improve sensitivity and distinguishability of the ROIs [30]. Others have focused on demixing the recorded signals, instead of localizing them, after which their connectivity is estimated, e.g., using the method proposed in [16].

We proposed Sparse eLORETA for source localization. It relies on Structured Sparse Bayesian Learning (SSBL) to reduce source space dimensionality. Combined with eLORETA, it yields a much sparser solution, while retaining

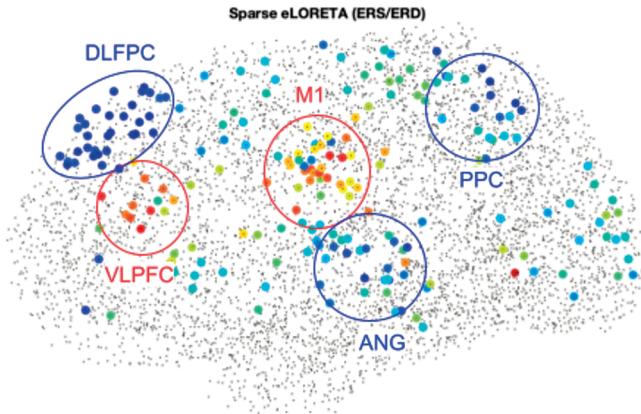


Figure 2.5: Contralateral view of the Sparse eLORETA solution with clusters labeled by the corresponding brain regions.

the advantages of the latter (i.e., a low localization error and few ghost sources). The much sparser solution is expected to facilitate subsequent connectivity analysis.

Similar to other simulation studies, we modulated the depth of the sources, their reciprocal distance, and the SNR (brain noise level). We adopted the framework of Anzolin et al. to develop a realistic simulation of brain activity to assess Sparse eLORETA’s performance for multiple configurations of the 3 active sources [16]. As expected, superficial sources that are far apart were much easier to localize, compared to deep sources close together, as seen in Table 2.3. Compared to LCMV and eLORETA, Sparse eLORETA performed better, gaining more sparsity while sacrificing only minimally on accuracy, showing the benefit of SSBL. We mainly looked at how accurately these state-of-the-art techniques could locate 3 active sources. We had instances where the sources were not accurately located but their signals correlated well with the true ones. Moreover, we could show that, despite the SSBL step, the computational effort of our method is similar to eLORETA’s since our solution is more sparse. However, BC-VARETA requires significantly more effort compared to eLORETA and our method. LCMV is the fastest to compute, although the results were disappointing.

To quantify source localization accuracy of our simulation results, we performed a medoid clustering analysis on the reconstructed source distributions. We extracted two metrics from this analysis, the distance of the medoids to the

true sources and the spatial spread around the former, and showed that our method performed better than eLORETA. However, our results also showed that BC-VARETA still holds an edge over the methodology we propose as seen in Table 2.5. Since BC-VARETA localizes sources but does not reconstruct their time series, a comparison with BC-VARETA cannot account for the fidelity with which these time series are reconstructed. In that sense, our method is a compromise between source sparsity and signal fidelity, two aspects of the source reconstruction problem. Medoid clustering could also be used to reduce source space similar to Wang et al. [31], albeit using different clustering approaches. We believe that source screening, which we proposed for eLORETA, and (data-driven) clustering, or a combination thereof, have their merits in generating sparse solutions.

In order to show the applicability in practice, we considered a resting state and a cued imagined movement EEG dataset. For the first case, except for LCMV, we observe that the source estimates of eLORETA, Sparse eLORETA and BC-VARETA are located in similar regions but with a smaller spread for the latter two, as expected (Fig. 2.3). However, while in our simulation study BC-VARETA exhibited a smaller spread, this was no longer the case with the real-world dataset.

As to the results of the second case (Figure 2.4), we observe that, compared to Sparse eLORETA, eLORETA has more ipsilateral clusters of sources but at the same time more spurious ones. Again, BC-VARETA performs much worse by consistently locating sources in medial central and temporal cortices (with almost zero spread).

Finally, for cases where ROIs are not predefined, such as with the real world datasets we have also considered, our method could be used to perform source space reduction prior to applying source localization. In the future, it would be interesting to expand on our work by integrating Sparse eLORETA with a method for selecting ROIs. In this way, we could arrive at a completely data-driven method. Since source localization results returned by Sparse eLORETA are sparse, it would be interesting to see how clustering algorithms based on pairwise connectivities could assist in charting ROIs.

2.6 Conclusion

We show that Sparse eLORETA offers the possibility for eLORETA to trade temporal for spatial accuracy.

2.7 Funding

AF was supported by the Belgian Fund for Scientific Research – Flanders (FWO 1157019N). AdB participated as part of an internship. MMVH is supported by research grants received from the European Union’s Horizon 2020 research and innovation programme onder grant agreement No.857375, the Financing Program (PFV/10/008) and the special research fund of the KU Leuven (C24/18/098), the Belgian Fund for Scientific Research – Flanders (G088314N, G0A0914N, G0A4118N), the Interuniversity Attraction Poles Programme – Belgian Science Policy (IUAP P7/11), and the Hercules Foundation (AKUL 043).

Bibliography

- [1] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.
- [2] M. X. Cohen, *Analyzing neural time series data: theory and practice*. MIT press, 2014.
- [3] C. M. Michel and D. Brunet, “Eeg source imaging: A practical review of the analysis steps”, *Frontiers in neurology*, vol. 10, p. 325, 2019.
- [4] J.-M. Schoffelen and J. Gross, “Source connectivity analysis with meg and eeg”, *Human brain mapping*, vol. 30, no. 6, pp. 1857–1865, 2009.
- [5] R. D. Pascual-Marqui, “Review of methods for solving the eeg inverse problem”, *International journal of bioelectromagnetism*, vol. 1, no. 1, pp. 75–86, 1999.
- [6] R. Grech, T. Cassar, J. Muscat, K. P. Camilleri, S. G. Fabri, M. Zervakis, P. Xanthopoulos, V. Sakkalis, and B. Vanrumste, “Review on solving the inverse problem in eeg source analysis”, *Journal of neuroengineering and rehabilitation*, vol. 5, no. 1, pp. 1–33, 2008.
- [7] B. D. Van Veen, W. Van Drongelen, M. Yuchtman, and A. Suzuki, “Localization of brain electrical activity via linearly constrained minimum variance spatial filtering”, *IEEE Transactions on biomedical engineering*, vol. 44, no. 9, pp. 867–880, 1997.
- [8] K. Whittingstall, G. Stroink, L. Gates, J. Connolly, and A. Finley, “Effects of dipole position, orientation and noise on the accuracy of eeg source localization”, *Biomedical engineering online*, vol. 2, no. 1, pp. 1–5, 2003.
- [9] R. D. Pascual-Marqui, “Discrete, 3d distributed, linear imaging methods of electric neuronal activity. part 1: Exact, zero error localization”, *arXiv preprint arXiv:0710.3341*, 2007.
- [10] R. D. Pascual-Marqui, D. Lehmann, M. Koukkou, K. Kochi, P. Anderer, B. Saletu, H. Tanaka, K. Hirata, E. R. John, L. Prichep, et al., “Assessing interactions in the brain with exact low-resolution electromagnetic tomography”, *Philosophical Transactions of the Royal*

- Society A: Mathematical, Physical and Engineering Sciences*, vol. 369, no. 1952, pp. 3768–3784, 2011.
- [11] E. Gonzalez-Moreira, D. Paz-Linares, E. Martinez-Montes, and P. A. Valdes-Sosa, “Third generation meeg source connectivity analysis toolbox (bc-vareta 1.0) and validation benchmark”, *arXiv preprint arXiv:1810.11212*, 2018.
 - [12] D. Paz-Linares, E. Gonzalez-Moreira, A. Duru, and P. Sosa, “Measuring and suppressing m/eeg connectivity leakage: The hidden gaussian graphical state-model (higgs)”, Jun. 2019. doi: 10.13140/RG.2.2.15536.99846.
 - [13] M. L. Bringas Vega, Y. Guo, Q. Tang, F. A. Razzaq, A. Calzada Reyes, P. Ren, D. Paz Linares, L. Galan Garcia, A. G. Rabinowitz, J. R. Galler, *et al.*, “An age-adjusted eeg source classifier accurately detects school-aged barbadian children that had protein energy malnutrition in the first year of life”, *Frontiers in neuroscience*, vol. 13, p. 1222, 2019.
 - [14] K. Rykaczewski, J. Nikadon, W. Duch, and T. Piotrowski, “Supfunsim: Spatial filtering toolbox for eeg”, *Neuroinformatics*, vol. 19, no. 1, pp. 107–125, 2021.
 - [15] D. Paz-Linares, M. Vega-Hernandez, P. A. Rojas-Lopez, P. A. Valdes-Hernandez, E. Martinez-Montes, and P. A. Valdes-Sosa, “Spatio temporal eeg source imaging with the hierarchical bayesian elastic net and elitist lasso models”, *Frontiers in neuroscience*, vol. 11, p. 635, 2017.
 - [16] A. Anzolin, P. Presti, F. Van De Steen, L. Astolfi, S. Haufe, and D. Marinazzo, “Quantifying the effect of demixing approaches on directed connectivity estimated between reconstructed eeg sources”, *Brain topography*, vol. 32, no. 4, pp. 655–674, 2019.
 - [17] S. Haufe and A. Ewald, “A simulation framework for benchmarking eeg-based brain connectivity estimation methodologies”, *Brain topography*, vol. 32, no. 4, pp. 625–642, 2019.
 - [18] A. Omidvarnia, M. Mesbah, J. M. O’Toole, P. Colditz, and B. Boashash, “Analysis of the time-varying cortical neural connectivity in the newborn eeg: A time-frequency approach”, in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, IEEE, 2011, pp. 179–182.
 - [19] S. Haufe, V. V. Nikulin, K.-R. Müller, and G. Nolte, “A critical assessment of connectivity measures for eeg data: A simulation study”, *Neuroimage*, vol. 64, pp. 120–133, 2013.
 - [20] E. A. Habets, I. Cohen, and S. Gannot, “Generating nonstationary multisensor signals under a spatial coherence constraint”, *The Journal of the Acoustical Society of America*, vol. 124, no. 5, pp. 2911–2917, 2008.

- [21] S. Haufe and A. Ewald, *Berlin brain connectivity benchmark and data analysis challenge*, <http://bbci.de/supplementary/EEGconnectivity/BBCB.html>, 2016.
- [22] Y. Huang, L. C. Parra, and S. Haufe, “The new york head—a precise standardized volume conductor model for eeg source localization and tes targeting”, *NeuroImage*, vol. 140, pp. 150–162, 2016.
- [23] G. Hernandez-Gonzalez, M. L. Bringas-Vega, L. Galán-Garcia, J. Bosch-Bayard, Y. Lorenzo-Ceballos, L. Melie-Garcia, L. Valdes-Urrutia, M. Cobas-Ruiz, P. A. Valdes-Sosa, and C. H. B. M. P. (CHBMP), “Multimodal quantitative neuroimaging databases and methods: The cuban human brain mapping project”, *Clinical EEG and neuroscience*, vol. 42, no. 3, pp. 149–159, 2011.
- [24] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Muller, “Boosting bit rates in noninvasive eeg single-trial classifications by feature combination and multiclass paradigms”, *IEEE transactions on biomedical engineering*, vol. 51, no. 6, pp. 993–1002, 2004.
- [25] J. Munzert, B. Lorey, and K. Zentgraf, “Cognitive motor processes: The role of motor imagery in the study of motor representations”, *Brain research reviews*, vol. 60, no. 2, pp. 306–326, 2009.
- [26] M.-P. Deiber, V. Ibanez, M. Honda, N. Sadato, R. Raman, and M. Hallett, “Cerebral processes related to visuomotor imagery and generation of simple finger movements studied with positron emission tomography”, *Neuroimage*, vol. 7, no. 2, pp. 73–85, 1998.
- [27] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.
- [28] D. Mantini and W. Vanduffel, “Emerging roles of the brain’s default network”, *The Neuroscientist*, vol. 19, no. 1, pp. 76–87, 2013.
- [29] M. Fahimi Hnazaee, E. Khachatryan, S. Chehraziad, A. Kotarcic, M. De Letter, and M. M. Van Hulle, “Overlapping connectivity patterns during semantic processing of abstract and concrete words revealed with multivariate granger causality analysis”, *Scientific reports*, vol. 10, no. 1, pp. 1–19, 2020.
- [30] S.-R. Farahibozorg, R. N. Henson, and O. Hauk, “Adaptive cortical parcellations for source reconstructed eeg/meg connectomes”, *NeuroImage*, vol. 169, pp. 23–45, 2018.
- [31] S. H. Wang, M. Lobier, F. Siebenhühner, T. Puoliväli, S. Palva, and J. M. Palva, “Hyperedge bundling: A practical solution to spurious interactions in meg/eeg source connectivity analyses”, *NeuroImage*, vol. 173, pp. 610–622, 2018.

Chapter 3

Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data

AXEL FAES, IRIS VANTIEGHEM AND MARC M. VAN HULLE

Applied Sciences 12.6 (2022): 2889.

3.1 Abstract

Directed connectivity between brain sources identified from scalp electroencephalography (EEG) can shed light on the brain's information flows and provide a biomarker of neurological disorders. However, as volume conductance results in scalp activity being a mix of activities originating from multiple sources, the correct interpretation of their connectivity is a formidable challenge despite source localization being applied with some success. Traditional connectivity approaches rely on statistical assumptions that usually do not hold for EEG, calling for a model-free approach. We investigated several types of Artificial Neural Networks in estimating Directed Connectivity between Reconstructed EEG Sources and assessed their accuracy with respect to several ground truths. We show that a Long Short-Term Memory neural network with Non-Uniform

Embedding yields the most promising results due to its relative robustness to differing dipole locations. We conclude that certain network architectures can compete with the already established methods for brain connectivity analysis.

3.2 Introduction

A challenging problem in neuroimaging is to estimate directed connectivity between brain regions reconstructed from scalp EEG recordings but important to unveil their joint dynamics. Due to volume conduction, a given EEG electrode can pick up signals from several sources simultaneously, distorted along the way due to the presence of tissues with different electrical properties. Resolving these sources is called the "inverse problem", and it consists of estimating the source parameters given the scalp EEG recordings. The number of sources is higher than the number of electrodes, rendering an ill-posed problem. Valid brain connectivity estimation critically depends on the correct localization and time series reconstruction in this stage. Several localization methods have been proposed, often yielding differing outcomes. In a comprehensive set of simulations, [1] studied the influence of several inverse solutions, the depth of the sources, their reciprocal distance, and the Signal-to-Noise Ratio (SNR) of the recordings. They found that all these factors had a significant impact on the resulting connectivity pattern and that the number of spurious connectivity estimations depends heavily on the combinations of these factors.

In addition to the said factors, the choice of the connectivity estimator also has a significant impact. Our interest lies in directed connectivity estimation, of which partial directed coherence [2], dynamic causal modeling [3], structural equation modeling [4] and (conditional) Granger causality (GC) [5] are well-known methods. However, they rely on statistical assumptions that usually do not hold for EEG data, such as linearity [6], stationarity and prior assumptions on connectivity being expressible as a relation between time series. However, even though some of these assumptions are violated, these methods still are best practice cases of directed connectivity estimation. In what follows, we focused on variations in traditional Granger Causality, given that it does not rely on an a priori assumed connectivity pattern. Granger Causality is a statistical hypothesis used to determine temporal causal effects between two time series. If the past of a second time series (Z) together with the past of a first time series (Y) (i.e., the "full" model) results in an improved prediction of the future value of the first time series, than the past of the first time series alone (the "reduced" model), it is said that time series Z "Granger-causes" Y .

Two main problems with this bivariate model can be discerned. Firstly, bivariate GC does not account for other time series that may be causing both Y and

Z , resulting in spurious connectivity patterns. Secondly, even when bivariate GC is extended towards multiple time series by conditioning on these other variables, it is still possible that the found influence is actually caused by a linear mixture of non-interacting sources. This is because the signal measured from one electrode usually contains contributions of several sources [7]. Important to note is the proposal of Time-Reversed Granger Causality (TRGC) by [8], further validated by [7], to reduce the impact of additive correlational noise due to source mixing. The idea is that when connectivity is based on temporal delay, directed connectivity should be reversed when the temporal order is reversed. Concretely, it is checked whether the obtained GC scores for non-reversed and reversed data have opposing directions and are both significant [1]. This is clearly different from a classical way to determine significance (i.e., a likelihood ratio test). Hence, the main difference between TRGC and traditional GC is the proposed significance procedure. Still, even with TRGC, errors in connectivity estimation are here to stay. The question remains whether a totally different approach could cope with the above-mentioned problems and could perform better, or at least equally well, in comparison with the standard approaches. Artificial Neural Networks (ANNs) were considered as particularly interesting candidates given their flexible way of approximating highly non-linear relationships between variables [9] and the fact that no a priori assumptions need to be made about signal stationarity nor the connectivity pattern (for a clear overview, see [10]). Temporal convolutional networks (TCNs), as well as recurrent neural networks (RNNs), are usually well-suited architectures for time series [11]–[15]. While RNNs are often seen as the gold standard for sequence modeling, TCNs have also proven their suitability, for instance, in financial forecasting [14], electric power forecasting [16] and language modeling [17]. However, it remains unclear whether ANNs can signal the presence or absence of connections and their strength. Although some authors already used ANNs to derive directed brain connectivity with multilayer perceptrons and recurrent networks [15], [18], these approaches did not include source-reconstructed EEG data. As stated before, unlike EEG source reconstruction, analyses based on EEG electrode levels do not allow for trustworthy inferences about interacting regions [19]. Hence, the suitability of ANNs in deriving directed connectivity between reconstructed EEG sources remains unknown.

Our motivation to assess ANNs for directed connectivity estimation between reconstructed EEG sources was two-fold. First, although many connectivity estimators exist, it is not yet known which current ANNs architectures can cope better with source-reconstructed EEG activity and under various circumstances. The authors of [1] were the first to conduct a comprehensive simulation study on the influence of dipole location, noise level, inverse solution and connectivity estimation, as well as the interactions between these factors. It was shown that different circumstances call for different analysis pipelines and that

under advanced noise levels and for particular dipole configurations, even well-established methods such as TRGC can return aberrant connectivity estimates. Second, ANNs boast several appealing modeling properties that are potentially relevant to EEG modelers, such as the ability to deal with non-stationarity, non-linearity and, depending on the ANN architecture, to dispense with the prior specification of model order.

In order to assess the ability of ANNs to correctly signal the presence or absence of directed connectivity as well as connectivity strength, we compared several ANN models, including Conv2D, a novel ANN model we propose, with TRGC. We compared their performance for different dipole locations (i.e., Far-Deep/Far-Superficial) as this can inform us whether there is a future for ANN models in brain connectivity estimation. In addition, we evaluated the ANN models relevant for directed connectivity estimation. We investigated these issues by means of a simulation study, thereby making use of a slightly adapted version of the simulation framework developed by [1] in which we manipulated the location of the dipoles and their connectivity while keeping noise level and the choice of the inverse solution constant.

3.3 Materials and Methods

3.3.1 Simulation Procedure

The simulation framework developed by [1] was used to generate simulated EEG data originating from three dipoles. This data generating process, as well as the forward and inverse problems, were implemented in MATLAB (2020). Figure 3.1 shows the data generation procedure. The standard length of each generated series was 1500 time steps for Ground Truth 1 and 2.

In Ground Truth 1, three fixed dipoles were used with the directionality of the connections as well as their strength being imposed (Figure 3.2), a strategy used before [20]–[22]:

$$\begin{cases} X_1(n) = 0.5X_1(t-1) - 0.7X_1(t-2) \\ \quad + c_{12}(t)X_2(t-1) + w_1(t) \\ X_2(n) = 0.7X_2(t-1) - 0.5X_2(t-2) \\ \quad + 0.2X_1(t-1) + c_{23}(t)X_3(t-1) + w_2(t) \\ X_3(n) = 0.8X_3(t-1) + w_3(t) \end{cases} \quad (3.1)$$

with X_1 , X_2 and X_3 , three electrical sources contributing to the simulated scalp-EEG signals and with:

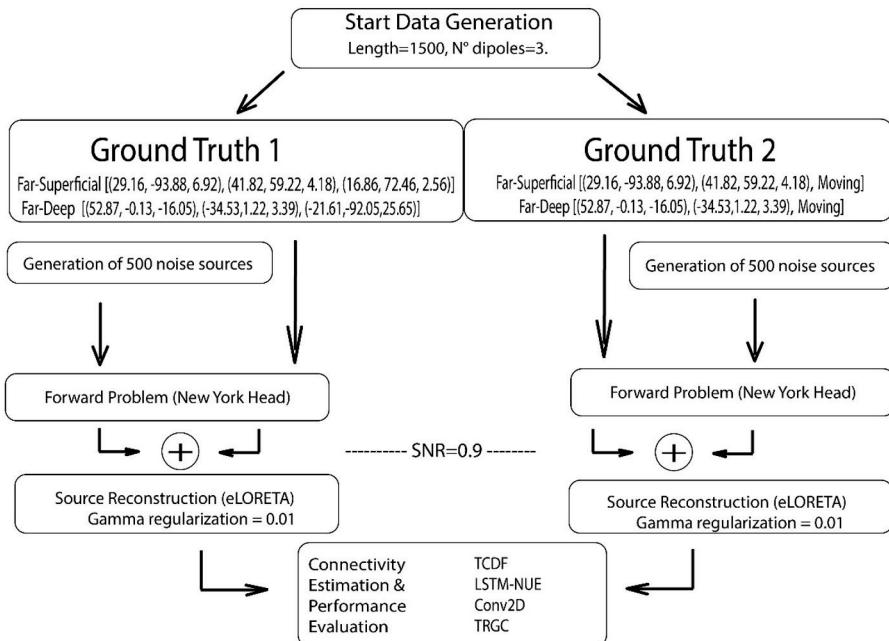


Figure 3.1: Simulation Procedure followed by Connectivity Estimation. TCDF = Depthwise Separable 1D Temporal Causal Discovery Framework, LSTM-NUE = Long Short-Term Memory with Non-Uniform Embedding, Conv2D = 2D Convolutional Network, TRGC = Time-Reversed Granger Causality. Coordinates in the Ground Truths denote MNI-coordinates.

$$\begin{cases} c_{12} = 0.5 \frac{t}{L} & \text{if } t \leq \frac{L}{2} \\ c_{12} = 0.5 \frac{L-t}{\frac{L}{2}} & \text{if } t > \frac{L}{2} \\ c_{23} = 0.4 & \text{if } t < 0.7L \\ c_{23} = 0 & \text{if } t \geq 0.7L \end{cases} \quad (3.2)$$

L = length of the generated time series ($L = 1500$), t = the current time step and w = uncorrelated white noise, varying with time. We further assume an EEG cap with 108 electrodes.

For Ground Truth 2, we considered two fixed, one moving dipole and only one true connection (Figure 3.3) and focused on the presence or absence of this connectivity as well as its directionality:

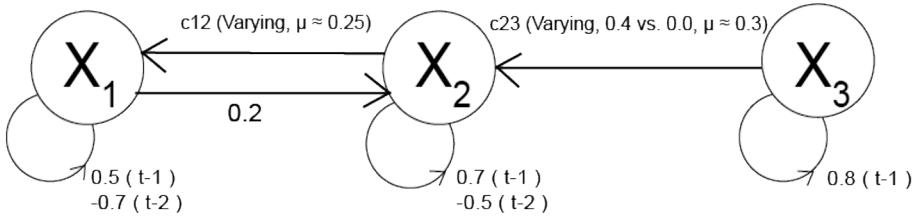


Figure 3.2: Ground Truth 1 with three fixed dipoles.

$$\begin{bmatrix} X_s(t) \\ X_r(t) \\ X_n(t) \end{bmatrix} = \sum_{p=1}^P \begin{bmatrix} a_{11}(p) & 0 & 0 \\ a_{21}(p) & a_{22}(p) & 0 \\ 0 & 0 & a_{33}(p) \end{bmatrix} \begin{bmatrix} X_s(t-p) \\ X_r(t-p) \\ X_n(t-p) \end{bmatrix} + \begin{bmatrix} w_s(t) \\ w_r(t) \\ w_n(t) \end{bmatrix} \quad (3.3)$$

with X_s , the moving dipole, as a sender, and two fixed dipoles, with X_r the receiver and X_n the fixed non-interactive dipole, and $a_{ij}(p)$, $i, j \in 1, 2, 3$ and $p \in 1, \dots, P$ the coefficients with a_{21} the coupling strength between sender and receiver. All a_{ij} are randomly picked from the interval $[0.3, 1]$. Finally, w is uncorrelated, biological, white noise.

The moving dipole (the sender) changes location (far, deep, close, superficial) at every iteration, with a total of 1004 iterations. The maximum time lag t is two. The reason for this ground truth is that the sender can be located at really challenging locations (too close to one of the other dipoles or very deep in the brain).

Two conditions were created for both ground truths: one condition consisted of three superficial dipoles far away from each other, while the other consisted of three dipoles located "deep" in the brain, but each dipole was still positioned far away from the other dipoles. The corresponding MNI-coordinates of the two fixed dipoles that Ground Truth 1 and 2 have in common are depicted in Figure 3.4. The full set of coordinates of Ground Truth 1 (including the coordinates of the third fixed dipole) is denoted in Figure 3.2.

In Ground Truth 2, the first two coordinates are the same as in Ground Truth 1, for each dipole condition, while the third dipole moves throughout the brain as described above. The Far-Superficial versus Far-Deep configurations indicate (relative) distances: "deep" denotes a distance from the origin (located at the anterior commissure) < 6 cm and "superficial" > 6.5 cm. The distance between dipoles is evaluated as "far" if the relative distance to the other dipoles exceeds 8 cm.

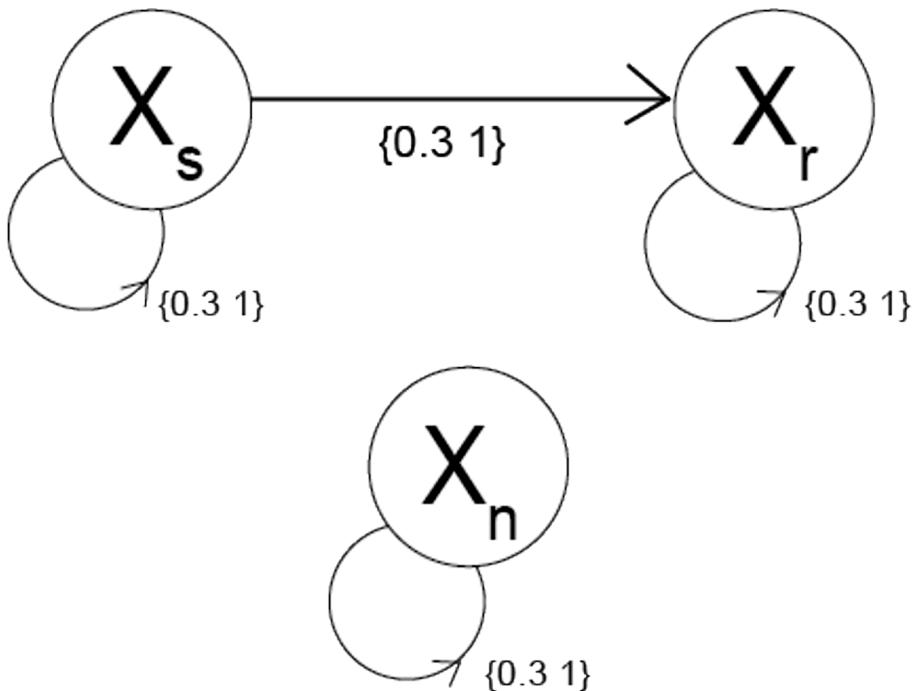


Figure 3.3: Ground Truth 2 with two fixed, one moving dipole.

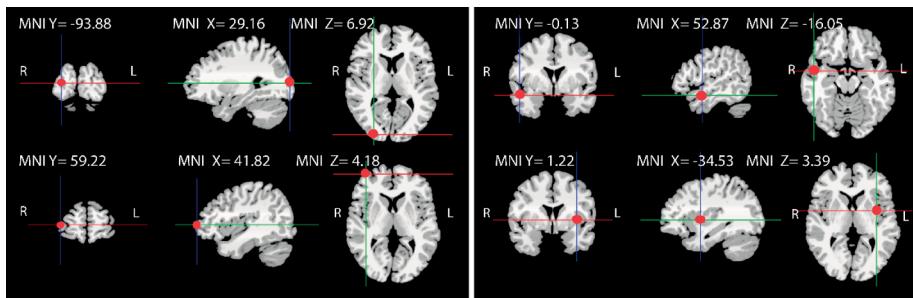


Figure 3.4: Locations (MNI-coordinates) of the dipoles the Ground Truths have in common. Left: Far-Superficial fixed dipoles. Right: Far-Deep fixed dipoles.

As an additional check for robustness of source localization, noise sources were added as a background activity. These were modeled using pink noise, also called $1/f$ noise, and created by scaling the amplitude spectrum of random

white Gaussian noise with the factor 1/f using the Fourier transform and its inverse.

After generating these noise sources, the forward problem is construed:

$$Y = LX + e \quad (3.4)$$

where Y denotes the scalp-recorded potentials, X represents the electrical sources in the brain (the dipoles), "e" is measurement noise (electrode noise) and L is the head volume conductor model (also called the leadfield matrix). The leadfield matrix determines how the activity flows from dipoles to electrodes. In this work, the New York Head model [23] was used.

The pink noise and the source activity are then projected onto the scalp, after which they are summed:

$$Y^{brain}(t) = \lambda \times \frac{Y^{active}(t)}{\|Y^{active}(t)\|_{FRO}} + (1 - \lambda) \times \frac{Y^{noise}(t)}{\|Y^{noise}(t)\|_{FRO}}. \quad (3.5)$$

Y^{active} and Y^{noise} refer to the scalp-projected source signals and pink noise activity, respectively; both are scaled by dividing them by their Frobenius norm ($\|Y^{active}(t)\|_{FRO}, \|Y^{noise}(t)\|_{FRO}$). The Signal-to-Noise Ratio (SNR) is computed for all dipoles simultaneously and set to 0.9 ($\lambda = 0.9$).

Next, white noise (spatially and temporally uncorrelated activity) is added to Y^{brain} to simulate electrode noise, resulting in Equation 3.6 where $Y^{measurement}$ represents the simulated EEG signal. Again $\lambda = 0.9$ is imposed as Signal-to-Noise Ratio:

$$Y^{measurement}(t) = 0.9 \times \frac{Y^{brain}(t)}{\|Y^{brain}(t)\|_{FRO}} + 0.1 \times \frac{Y^{m_noise}(t)}{\|Y^{m_noise}(t)\|_{FRO}}. \quad (3.6)$$

Afterwards, the simulated scalp-EEG data are source-reconstructed using exact low-resolution brain electromagnetic tomography (eLORETA) [24]. There have also been improvements to eLORETA, such as Sparse eLORETA, which uses a masking approach to improve the source localization density [25]. The eLORETA method is a discrete, three-dimensional (3D), linear, weighted minimum norm inverse solution [24]. In the absence of noise, an exact zero-error localization accuracy can be obtained with eLORETA, but this does not hold for noisy data, as was shown in a study comparing both scenarios [26]. The MATLAB implementation of the eLORETA algorithm (mkfilt_eloreta2.m) from which spatial filters are obtained was developed by

Model	Architecture	Self-Causation	Connectivity Measure
TCDF	Depthwise Separable 1D Convolutional Network	yes	Attention score
LSTM-NUE	Long Short-Term Memory Network	no	GC (0 or 1)
Conv2D	2D Convolutional Network	yes	R2-score
TRGC	Time-Reversed Granger Causality	no	GC (0 or 1)

Table 3.1: Set-up of the compared ANNs and TRGC.

G. Nolte and is available in the MEG/EEG Toolbox of Hamburg (METH; <https://www.uke.de/english/departments-institutes/institutes/neurophysiology-and-pathophysiology/research/research-groups/index.html>, accessed on 22 December 2021). As the input, it takes the leadfield tensor (i.e., the head model file N^*M^*P containing N channels, M voxels, and P dipole directions) as well as a regularization parameter gamma (set to 0.01); as the output, an N^*M^*P tensor A of spatial filters is returned.

3.3.2 Connectivity Models

The ANNs evaluated in this study were selected based on their suitability for time series analysis (Table 3.1). While TCDF outputs attention scores, for which higher scores are used to represent stronger connectivities, LSTM-NUE makes use of Granger Causality scores equaling $NNGC = \text{errreduced-errfull}$, which are then binarized [18]. Conv2D uses the R2-score between the real and predicted values of the current target (i.e., the time steps to be predicted). While using TRGC as implemented by [1], only binary GC scores are outputted. The configuration (i.e., the used parameters) of each ANN was determined using a data-driven approach, such that for each ANN, the parameters returning the best results were chosen. This parameters pre-testing was performed with different simulated data sets (i.e., differing from the data sets that were used to report the final results).

Temporal Causal Discovery Framework

The Temporal Causal Discovery Framework (TCDF) developed by [14] is based on the concept of a one-dimensional Temporal Convolutional Network (TCN)

and is available on Github [27]. Input to the framework consists of an NxL data set consisting of N time series of equal Length L. Within the framework, one depthwise-separable TCN is used to obtain a prediction for a single source (target). The input of the network consists of the history of all time series, including the target time series. The output is the history of the target time series. An attention mechanism is added: each TCN_j has its own trainable attention vector $V_j = [vX_{1j}, vX_{2j}, \dots, v_{ij}, \dots, vN_{jj}]$, that learns which of the input time series is correlated with the target by multiplying attention score v_{ij} with input time series X_i in TCN_j. When the training of the network starts, all attention scores are initialized as 1 and are, as such, adapted during training. The direction of connectivity and significance is determined using a shuffling procedure. For significance determination, one of the time series is shuffled while keeping the other one(s) intact when predicting the target. The runs with shuffled time series did not involve any model retraining. Instead, in the prediction step, the losses obtained when using the “shuffled” time series as predictors were compared with the losses obtained when using the non-shuffled time series. Only if the loss of a network increases significantly when a time series is shuffled that time series is considered a cause of the target time series. A time series X_1 is only considered to be a significant contributor to another time series X_2 if, in the first stage, its attention score is larger than one. Only if, after shuffling the potentially contributing time series X_1 , the difference between losses obtained by predicting future time steps with the unshuffled time series and losses obtained by predicting using shuffled time series is large enough, using an a priori determined threshold significance value, time series X_1 is considered a significant contributor to time series X_2 . TCDF was run with PyTorch (version 1.4.0, www.pytorch.org, accessed on 17 December 2021).

Configuration. For TCDF, the chosen parameters were the number of hidden layers = 1, kernel size = dilation coefficient = 4 (a time-dimensional kernel), learning rate = 0.01, optimizer = Adam, number of epochs = 1000, significance threshold= 0.9998, seed = 1000. Kernel weights are initialized following a distribution with $\mu = 0$, variance = 0.1.

LSTM-NUE—Long Short-Term Memory with Non-Uniform Embedding

Another connectivity measure is based on the RNN, in which directed cycling connections are present, i.e., there are feedback connections from output to input, and these connections create possibilities for memorization. A subtype of RNNs is the Long Short-Term Memory network (LSTM). This type of network provides a resolution for vanishing and exploding gradient problems in recurrent networks. It performs this by introducing gates and memory cells which also makes it very flexible towards gap length. The implementation in this study is an LSTM with Non-Uniform Embedding (NUE, a feature selection procedure)

by [15], which is also publicly available [28]. NUE is an iterative selection procedure adopted from [18] to detect the most informative time steps of the predicting time series (phase one). In phase one, a vector V containing the most informative past time steps to explain the present state of a target time series X_1 is obtained by iteratively adding time steps (of the time series' own past, but also of the past of the other time series) to the training set and obtaining a new model error as a time step is added. For instance, let $V = [V^{X1n}, V^{X2n}, V^{X3n}]$ represent the vector with the most relevant past time steps to explain the present of the target time series. This selection of time steps goes on until the prediction error becomes larger than or equal to a threshold or until the maximum amount of time steps is reached. If for a certain time series X_2 , no time steps have been added in V , the time series is not further considered as a potential contributor to target time series X_1 , and it is not considered in the next phase (phase two). Phase one results in an estimation of the error variance of the full model (i.e., the model containing all relevant past time steps from different time series). In phase two, the model is fit only with this smaller set of time steps. The error of the reduced model is finally obtained by not using the values of the time series (e.g., X_3) that is a potential contributor to the target time series X_1 . If the error ($\text{Loss}^{\text{Reduced}}$) of this reduced model is larger than the error of the full model ($\text{Loss}^{\text{Full}}$), time series X_3 is considered a significant contributor to time series X_1 (" X_3 Granger-predicts X_1 ").

In LSTM-NUE, no shuffling is used to determine connectivity. Instead, the significance procedure consists of two phases. Determining significance is based on (1) the selection of relevant time samples from all time series rendering a full model, after which the time series whose time samples were not selected are already as potential causes of the target time series. (2) The remaining candidates are then, as a test, subsequently excluded from the model to obtain the reduced model (i.e., the model with only the target time series as its own predictor). Hence, this exclusion phase is, to some extent, comparable with the shuffling procedure used in TCDF, given that this procedure is in this way testing the relevance of a certain time series in the prediction of another (by excluding it OR by shuffling the values).

Configuration: for LSTM-NUE, the parameters are the number of hidden layers = 1, the number of units in each layer = 30, batch size = 30, num_shift = 1, sequence_length = 20, number of epochs = 100, theta = 0.09, learning rate = 0.001, weight decay = 1×10^{-7} , min_error = 1×10^{-7} (= a priori determined error to determine whether a certain time step should be included in the final model), and train/validation split = 0.85/0.15. Default kernel initializer = "glorot_uniform", which draws samples from a uniform distribution, is used to initialize the weights of the LSTM-layer.

Conv2D—Two-Dimensional Convolutional Network

Finally, we propose a two-dimensional Convolutional Network (Conv2D) as a way to test whether a 2D kernel variation in TCDF has merit. The input consists of an NxL data set, which is transformed into a four-dimensional tensor (time samples of training set, window size, amount of predicting time series, 1). The source code is accessible via Github ([kul-EEG-sourceconnectivity](https://github.com/irisv440/kul-EEG-sourceconnectivity), <https://github.com/irisv440/kul-EEG-sourceconnectivity>).

Some important differences with TCDF are the fact that a two-dimensional kernel is used and that a cross-validation procedure, adapted for time series, is embedded in the framework. While in TCDF, a one-dimensional kernel (with height = 1) slides over the data along the time dimension (= width of the kernel, i.e., the amount of time steps considered together), in Conv2D, a two-dimensional kernel is used in which the second dimension represents the amount of time series that will be convolved together. The second dimension has an upper bound, which is the total amount of time series within the input data. We hypothesized that by adding a second dimension (feature dimension) to TCN, we could capture the most important aspects of the other time series, leading to more correct connectivity estimates. However, it was suggested (e.g., [29]) that convolving data from several time series can also cause less accurate results (in our case, this means lower Sensitivity and lower Precision) because too many time series are convolved together, possibly erasing the impact of changes in individual time series. Similar to TCDF, the input to the network consists of all time series, including the target time series. The output is a single target time series.

A second difference is cross-validation (CV) for time series. Cross-validation is a powerful method for detecting overfitting, but its implementation in time series models is not trivial, given that no leakage from future to past may exist. This issue was solved by using 6-fold cross-validation on a rolling basis based upon “TimeSeriesSplit” from the model selection module of the sklearn-library version 0.24.1 (Scikit-learn, original version released by [30]). With TimeSeriesSplit, we obtained the following train-test regime for the folds where “—“represents the unused part of the data in the corresponding fold (Figure 3.5).

In addition, given that connectivity may vary over longer time spans (as is also the case in Ground Truth 1), working with only one division in the train/validation/test-set (respecting past versus future) can cause false positives or false negatives since one may be training on a portion of the time series where connectivity is very strong between, for instance, X3 and X2 while validating and/or testing on a part where the same connectivity is weak (or the other way around).

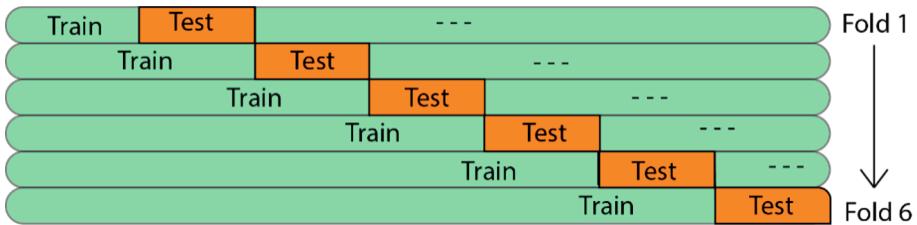


Figure 3.5: CV with length of first train-fold = length of first test-fold (= $1500/7$).

As a metric for connectivity strength, the R₂-score between the real values and the predicted values of the current target is used. The better a time series pair is successful in predicting a target, the larger the similarity between the true values and the predicted values will be, hence the stronger the connectivity between the time series and target. When, for instance, two different pairs of time series X₁ and X₂, versus X₁ and X₃ are used as predictors for X₁, R₂ again represents the similarity between predicted and true values of the target X₁. When the prediction of X₁ becomes better when predicted by time series X₁ and X₂ together, instead of with X₁ and X₃, one could conclude that connectivity is stronger between X₁ and X₂ than between X₁ and X₃. The R₂ scores themselves are obtained from the cross-validation folds, after which the average R₂ score is taken over the folds and over the number of used runs for one data set. The corresponding output is a scoring matrix representing all combinations of time series used as predictors and possible target time series. If the R₂ score is >0 and the predicting time series are considered significant (see “Connectivity Analysis using ANNs”), the obtained R₂ score can be interpreted. However, when including more than two predictors, this relationship is not so easily established anymore, given that the R₂-score still represents the connection between the target and all predicting time series together. Similar to TCDF, the direction of connectivity and significance is determined using a shuffling procedure. Significance weights are obtained by comparing training and test loss differences, after which a data-driven cutoff (here 0.70) is used to differentiate between contributing and non-contributing time series. More concretely, training difference = (first training loss)-(final training loss), where the latter is expected to be much lower than the first term, and Test difference = (first training loss)-(loss of test-indices using shuffled train data) where the latter is expected to be high because of the shuffled data; hence, one expects the test difference to be very small. Next, if the average test difference was larger than the average_training_difference * significance ($= 0.9998$), the potential connection is considered not significant in the first place. Significance weights are obtained by (test difference/training difference).

If the weight is larger than the cutoff ($= 0.70$), the connection is considered not significant. The used significance level, as well as the cutoff for significance weights, were experimentally determined, and the final choice was based upon a data-driven approach (by experimenting with significance levels in the range of 0.70, 1 and with cutoff-scores in the range of [0.40, 0.70]). For the current kind of simulated data, these values worked well.

Configuration: for Conv2D, the parameters were as follows: number of hidden layers = 1, number of filters = 24, kernel size = $4*2$, $4*3$ (width*height), dilation coefficient = 1, number of epochs = 12, window size = 5, learning rate = 0.005, optimizer = “Adam”, significance = 0.9998, cut-off scores for significance weights = 0.70 and number of train/test splits for CV = 6. Default kernel initializer = “glorot_uniform”, was used to initialize the weights of Keras’ Conv2D-layer.

TRGC—Time-Reversed Granger Causality

As our baseline method, Time-Reversed Granger Causality (TRGC), as implemented (by means of the Matlab function “`tr_gc_test`”, embedded in “`simulation_source_connectivity`”), and evaluated by [1], was used. As stated before, the difference with “traditional” GC is the type of significance procedure. Instead of the classical way to determine significance (a likelihood ratio test), which cannot distinguish between actual versus spurious correlations due to source mixing, it determines whether the “standard” GC scores for non-reversed and reversed data have opposing directions and are both significant. In other words, direction-flipping must occur when data are time-reversed. This is referred to as conjunction-based TRGC [7]. A drawback of GC (and hence, TRGC) is that one needs to define the model order, which is feasible when the ground truth is known, such as in simulations, but in “real” EEG data, this quickly becomes a tricky problem. An advantage, on the other hand, is the fact that with TRGC, one model for all sources is constructed, after which one threshold is applied to all obtained GC scores.

Configuration. Function `tr_gc_test` takes as input an $N \times L$ matrix H' , the model order, the number of time steps in the time series, alpha, the type of significance test (“conservative”, requiring significant GC scores with original as well as reversed data; versus a significance test based on difference scores between GC scores in normal and reversed order) and finally, the type of VAR model estimation regression mode to calculate pairwise-conditional time-domain Granger Causality scores. In this work, the model order of TRGC was set to two, we opted for “conservative” significance testing, and ordinary least squares (OLS) was used as Vector-Autoregression (VAR) estimator. We used an alpha level of 0.05, FDR corrected [31]. The corresponding p-value was taken as a threshold to binarize connectivity scores.

3.3.3 Performance Evaluation

The main question is whether the connections in the ground truths could be detected by the evaluated networks and by TRGC (“True Positives”, TP) without detecting too many false connections (“False Positives”, FP), thus connections that are not present in the ground truths. Measures based upon these are Precision, Sensitivity/Recall, and F1-score (Figure 3.6), which we used for comparing TCDF, LSTM-NUE, Conv2D and TRGC.

Connectivity-related measures:
1) Sensitivity (or Recall) = $\frac{TP}{TP+FN}$
2) Precision = $\frac{TP}{TP+FP}$
3) $F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$
General performance measure*:
Time Complexity (TC) = Runtime in seconds for five runs of one dataset on an Acer Aspire 7 A715-75G-751G, intel i7, 16 GB RAM.
*Used for comparison between ANNs only.

Figure 3.6: Main evaluation measures.

The results on connectivity strength are not directly compared between models as they differ substantially. These strength estimates, based on the mean over five runs on the same data set, are calculated and ranked. It must be emphasized that these strength estimates are relative per model and target training as, for each target time series, the network is trained differently. The latter implies that connection strengths obtained in the prediction of a particular Target time series X1 cannot be readily compared with connection strengths obtained in the prediction of another target time series X2. If $F1 < 50\%$, only rankings are presented. Self-connectivity is not taken into account to avoid an overly positive perception of the results.

3.4 Results

3.4.1 Ground Truth 1

Connectivity Detection

In Figures 3.7 and Figure 3.8, respectively, Sensitivity and Precision are shown per method. Some remarks, specifically with regard to Conv2D, need to be made before interpreting the results.

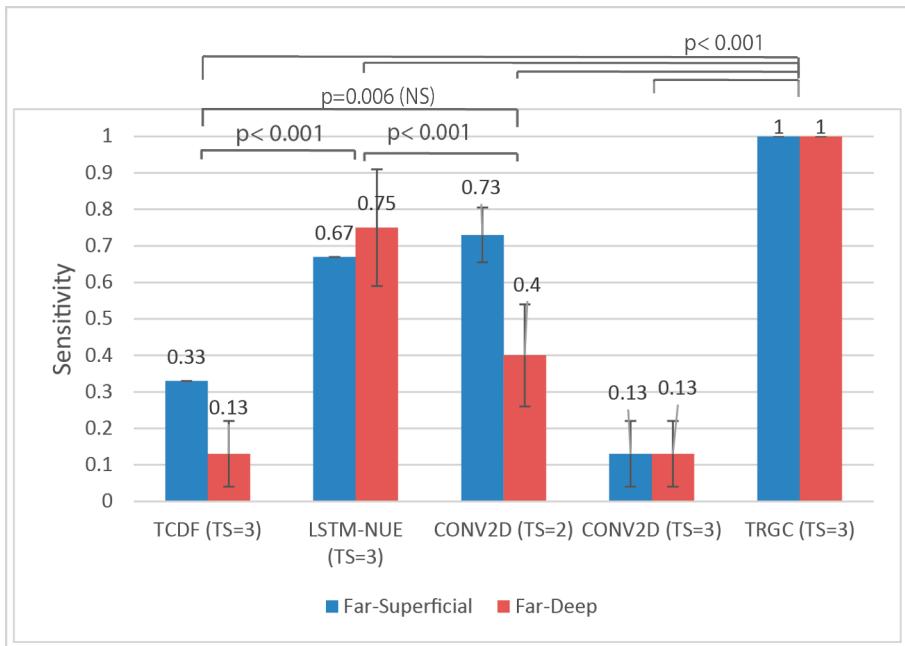


Figure 3.7: Mean Sensitivity for all methods ($L = 1500$). Sensitivity Ranking Far-Superficial: TRGC > Conv2D (TS = 2) > LSTM-NUE > TCDF > Conv2D (TS = 3). Sensitivity Ranking Far-Deep: TRGC > LSTM-NUE > Conv2D (TS = 2) > Conv2D (TS = 3) = TCDF. Abbreviation TS = amount of time series included in the predictions.

Given that the Conv2D-model based on three predicting time series resulted in a very low Sensitivity (0.13 ± 0.18) and low Precision (0.40 ± 0.55), see Figures 3.7 and Figure 3.8, it was not considered relevant to explore the model with three predictors further in terms of connectivity strength (for connectivity strength per ANN, see Sections 3.1.2 to 3.1.4). This decision was supported by the results of a Scheirer-Ray-Hare Test with model and dipole condition as factors and with follow-up Mann-Whitney U tests (Bonferroni-corrected).

Superior results were obtained with Conv2D models containing two predicting time series versus three predicting time series. These results can be consulted in Appendix B.

Hence, strength rankings are explored only with the Conv2D model with two predictors (Section 3.4.1). With regard to the model based on two predicting time series, the results obtained by looking at each predictor pair (consisting of two time series) separately revealed large differences between pairs in terms

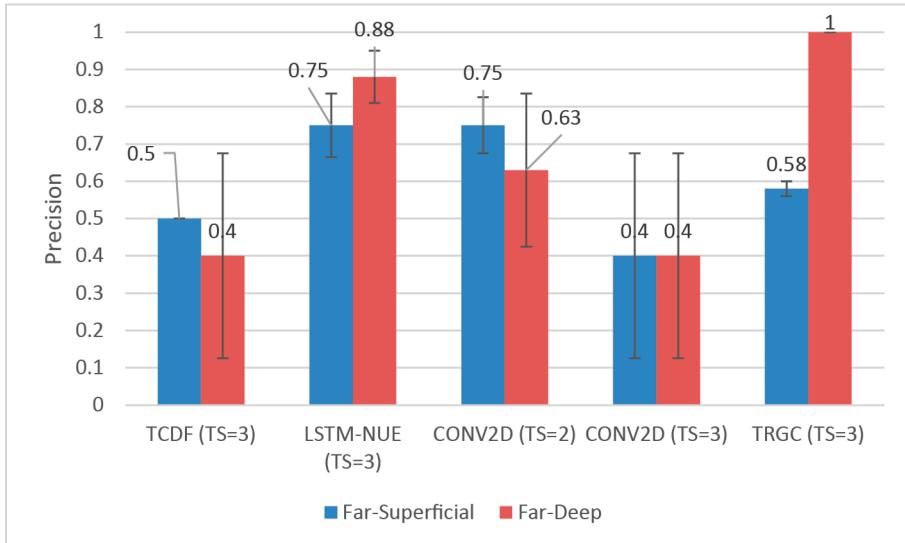


Figure 3.8: Mean Precision for all methods ($L = 1500$). Sensitivity Ranking Far-Superficial: TRGC > Conv2D (TS = 2) > LSTM-NUE > TCDF > Conv2D (TS = 3). Sensitivity Ranking Far-Deep: TRGC > LSTM-NUE > Conv2D (TS = 2) > Conv2D (TS = 3) = TCDF. Abbreviation TS = amount of time series included in the predictions.

of Sensitivity and Precision. We chose to take all detected connections into account while calculating our scores instead of averaging over all predictor pairs, as it could lead to biased results. This is because if it is found that X2 predicts X1 when it is predicted together with X1 but not detected when it is predicted with X2 and X3, and the discovered connection between X2 and X1 is still included in the performance scores, this increases Sensitivity but decreases Precision. The decrease in Precision then occurs because if a false positive is found by one of the two predictor pairs, it is still counted. Option 1 was chosen to put the focus more upon detection ability and exploration. Thus, it must be kept in mind that a positive detection bias exists in all our overall two-to-one performance scores of Conv2D.

While focusing on differences in Sensitivity, the following results were obtained for the used ANNs and TRGC. A Scheirer-Ray-Hare Test with model and dipole condition as factors revealed no statistically significant interaction (using alpha = 0.05) between the effects of the type of connectivity method and dipole condition ($p = 0.63$), nor the main effect of dipole condition itself ($p = 0.63$). However, a simple main effects analysis showed that the type of connectivity

method does have a statistically significant effect on Sensitivity ($H(4,40) = 38,159$, $p < 0.001$). Follow-up two-sided Mann-Whitney U tests (Bonferroni-corrected: alpha = 0.05, alpha adjusted = 0.005), carried out across dipole conditions, show significant and marginally significant differences between the following methods. Median scores (denoted as Mdn) are reported. In contrast to TCDF (Mdn = 0.33), smaller contributions of one time series to another could be detected with LSTM-NUE (Mdn = 0.67), $p < 0.001$. The difference between TCDF and Conv2D (Mdn = 0.67) with two time series as predictors was only marginally significant after correcting for multiple comparisons, $p = 0.006$. TRGC (Mdn = 1), however, outperformed all ANN models in terms of Sensitivity (p-values denoting differences with all other methods < 0.001). Finally, while no significant difference was found between LSTM-NUE (Mdn = 0.67) and Conv2D (Mdn = 0.67) with two time series as predictors ($p = 0.239$), LSTM-NUE performed significantly better than Conv2D with three time series as predictors (Mdn = 0), $p < 0.001$). Rankings are described below in Figure 3.7 to provide qualitative comparisons. Note that in Figure 3.7, mean scores M for each dipole condition is still reported, given that the current results were obtained with small sample sizes. Hence, differences between dipole conditions may still appear once statistical power is increased (i.e., by using more data sets) and given that differences between dipole conditions were, to some extent, expected.

With regard to Precision, the results of a Scheirer-Ray-Hare Test with method and dipole condition as factors were not significant, albeit a marginally significant result for method ($H(4,40) = 8.79$, $p = 0.07$) was obtained. Hence, no follow-up tests were carried out.

Thus, we rely upon rankings only for our qualitative description (in terms of mean scores M, taking dipole condition into account) of the data. In the Far-Superficial dipole condition, Conv2D with TS = 2 and LSTM-NUE obtain both a Precision of $M = 0.75 (\pm 0.15, 0.17$, respectively), followed by TRGC ($M = 0.58 \pm 0.05$) and TCDF ($M = 0.50 \pm 0$). Precision is lowest in Conv2D with TS = 3 ($M = 0.40 \pm 0.55$). However, in the Far-Deep dipole condition, TRGC obtains perfect Precision ($M = 1 \pm 0$), followed by LSTM-NUE ($M = 0.88 \pm 0.14$) and Conv2D with TS = 2 ($M = 0.63 \pm 0.41$). The qualitatively lower Precision score of TRGC in the Far-Superficial condition turned out to be mainly due to two consistently observed false-positive connections that were not detected in the Far-Deep dipole condition.

When summarizing the results in terms of F1-scores, the following ranking was obtained for the Far-Superficial condition: TRGC ($M = 0.73 \pm 0.04$) = Conv2D with TS = 2 ($M = 0.73 \pm 0.09$) $>$ LSTM-NUE ($M = 0.70 \pm 0.07$) $>$ TCDF ($M = 0.40 \pm 0.0$) $>$ Conv2D with TS = 3 ($M = 0.20 \pm 0.27$).

For the Far-Deep condition, the F1-score ranking was as follows: 1 ± 0 (TRGC) $> 0.75 \pm 0.17$ (LSTM-NUE) $> 0.47 \pm 0.31$ (Conv2D with TS = 2) $> 0.20 \pm 0.27$ (Conv2D with TS = 3) $= 0.20 \pm 0.27$ (TCDF).

TCDF

With regard to TCDF (Figure 3.9), only two mean attention scores were significant, and solely in the Far-Superficial dipole condition and for two different targets (X2 and X3), such that a target-wise comparison cannot be made. More concretely, a connection X3 \rightarrow X2 was found (in accordance with the ground truth), as well as a connection X2 \rightarrow X3 (unlike the ground truth).

	Far-Superficial			Far-Deep			Truth		
	T X1	T X2	T X3	T X1	T X2	T X3	T X1	T X2	T X3
X 1	-	NS	NS	-	NS	NS	-	3	NS
X 2	NS	-	1.07	NS	-	NS	2	-	NS
X 3	NS	1.43	-	NS	NS	-	NS	1	-

Figure 3.9: TCDF, Mean attention score rankings. Colors: “1”, green, denoting the highest attention score in one TCN—one column). The columns represent the targets (T X1, T X2, T X3), the rows the predictors (X1, X2, X3). Self-connectivity is excluded.

LSTM-NUE

Next, when focusing on LSTM-NUE, as can be seen from the colors from Figure 3.10, for target time series X3, the GC scores (in both dipole conditions) were higher than expected according to the ground truth. Unexpected GC scores are surrounded by black rectangles in the top panel. Column-wise strength rankings (rankings for one particular target) are correct for two out of three targets (X1, X2) in both conditions, as can be seen by comparing with column-wise Ground Truth 1 (Figure 3.10, bottom right panel). The overall ranking in the Far-Deep condition was more in accordance with the overall ranking in Ground Truth 1 (Figure 3.10, bottom left panel) than the ranking found in the Far-Superficial condition because connectivity strength was observed to be

the weakest for the corresponding false positives (as shown in yellow in the top panel).

Far-Superficial			Far-Deep			
	T X1	T X2	T X3	T X1	T X2	T X3
X1	-	5.50E-03 (\pm 1.10E-02)	NS	-	1.09E-02 (\pm 1.26E-02)	2.34E-03 (\pm 4.69E-03)
X2	3.78E-02 (\pm 1.48E-02)	-	3.20E-02 (\pm 3.32E-02)	3.08E-02 (\pm 3.66E-02)	-	2.75E-03 (\pm 5.51E-03)
X3	NS	1.19E-02 (\pm 8.98E-03)	-	NS	1.11E-01 (\pm 6.37E-02)	-

Truth 1: Overall ranking			Truth 1: Ranking per Column			
	T X1	T X2	T X3	T X1	T X2	T X3
X1	-	3	NS	-	2	NS
X2	2	-	NS	1	-	NS
X3	4	1	-	2	1	-

Figure 3.10: LSTM-NUE, Neural-Network Granger Scores rankings (Top) versus Truth 1 (Bottom), excluding self-connectivity. Color coding: dark green > light green > yellow > orange > red. The columns represent the targets, the rows the time series used for prediction.

Conv2D

In Figure 3.11, R2-strength rankings for Conv2D with two time series are shown. Given that for Conv2D, adding a third time series did not work out well, only rankings per predictor pair could be obtained. When R2-strength is shown in the upper two panels of Figure 3.11, it means that the current time series pair is a significant contributor. Significance weights, which denote significant contributions of one time series to a target time series (instead of R2 scores denoting a connection between a certain pair of predicting time series and one target time series), are reported between brackets. They were obtained as

described in Section 3.3.2 and were considered significant if the cutoff of 0.70 was not exceeded. The lower two panels show a ranking (with 1 being the most active connection and 4 the least active connection).

Far-Superficial			Far-Deep			
	T X1	T X2	T X3	T X1	T X2	T X3
X1, X2	0.35 (± 0.04)	0.31 (± 0.04)	NS	0.36 (± 0.05)	0.33 (± 0.04)	NS
X2, X3	NS	0.36 (± 0.04)	0.50 (± 0.04)	NS	0.36 (± 0.04)	0.51 (± 0.02)
X1, X3	0.34 (± 0.02)	0.02 (± 0.04)	0.49 (± 0.05)	0.34 (± 0.03)	0.02 (± 0.05)	0.51 (± 0.01)

Ground Truth 1: Overall ranking			Ground Truth 1: Ranking per column			
	T X1	T X2	T X3	T X1	T X2	T X3
X1, X2	2	3	NS	1	2	NS
X2, X3	4	2	1	2	1	1
X1, X3	3	4	1	3	3	1

Figure 3.11: Conv2D, R2-strength score for time series pairs (Top) versus ranking of connections in the Ground Truth 1 (Bottom, 1 being the strongest), including self-connectivity. Color coding: dark green > light green > yellow > orange > red. The columns represent the targets, the rows the time series pairs used for predicting the target time series.

It can be seen from Figure 3.11 that, in the prediction of Target X1, out of three direct connections, only one is not significant (i.e., X2, X3 \rightarrow X1, Top-row), but this is only the case when Target X1 is not included as a predictor. Regarding an overall ranking (Ground Truth 1, Bottom-Left), it can be seen that X3 has the strongest self-connectivity, while for X1 and X2, self-connectivity is almost the same. This is observed in our results as well ($R^2 = 0.36, 0.35$ in both dipole conditions). Moreover, the obtained R2 strength scores are not, or barely, dependent on the dipole condition. Next, while inspecting these results column-wise (hence, target-wise), a stronger connection between predictors X1, X2 and Target X1 than between predictors X1, X3 and target X1 were expected. However, these connections are quite similar ($R^2 = 0.35$

versus $R^2 = 0.36$ in the Far-Superficial condition, $R^2 = 0.36$ versus $R^2 = 0.34$ in the Far-Deep condition). While predicting target X2 using X1, X3, a significant, correct contribution from X1 to X2 is found (significance weight = 0.232, 0.120, Far-Superficial and Far-Deep condition, respectively), as well as a correct contribution from X3 to X2 (significance weight = 0.001, 0.048, Far-Superficial and Far-Deep condition, respectively). However, connectivity strength R^2 is very low ($R^2 = 0.02$ in both dipole conditions) in comparison with the situation in which target X2 is included in the predictor pair and in which case X3 is also considered a significant contributor (predictor pair = X2, X3, $R^2 = 0.36, 0.36$, significance weights = 0.341, 0.210 for X3, Far-Superficial and Far-Deep condition, respectively). The ranking for Target X2 is correct, as was the ranking for X1. Finally, we expected similar rankings for X2; X3 predicting X3 as for X1; X3 predicting X3 since neither X1 nor X2 contribute to X3. This is indeed the case for both conditions. As expected, significance weights for individual contributions of X1 and X2 to X3 were not significant (significance weights >0.70 in both dipole conditions).

Time Complexity

Finally, we assessed runtimes in seconds for one data set (including averaging over five runs) w.r.t. the training of the ANNs. Runtimes with three time series as predictors (TS = 3), Length L = 1500, dipole condition = Far-Superficial are shown in Figure 3.12, as well as the runtime of Conv2D with two time series as predictors. The runtime of Conv2D with only two time series as predictors was 1048 s. All runs were performed with an Acer Aspire 7 A715-75G-751G, intel i7, 16 GB RAM.

3.4.2 Ground Truth 2

With only one true connection and excluding self-connectivity, it was found that none of the methods, except for LSTM-NUE and TRGC (LSTM-NUE, TRGC: Sensitivity M = 1 ± 0 in both dipole conditions), were able to detect this connection in none of the runs or datasets (Table 3.2 and Table 3.3).

The results of a Scheirer-Ray-Hare Test with method and dipole condition as factors reveal, as expected, a main effect of connectivity method on Sensitivity ($H(4,40) = 47.66$, $p < 0.001$) as well as Precision ($H(4,40) = 48.10$, $p < 0.001$). The interaction between method and dipole condition, nor dipole condition itself were significant (Sensitivity: $H(4,40) = 0.05$, $p = 0.99$, $H(1,40) = 0.01$, $p = 0.91$, interaction and dipole condition effect, respectively; Precision: $H(4,40) = 0.03$, $p = 0.99$, $H(1,40) = 0.00$, $p = 0.95$, interaction and dipole condition effect, respectively). Looking into the effects of different connectivity methods using follow-up Mann-Whitney U tests (Bonferroni-corrected: alpha = 0.05, alpha

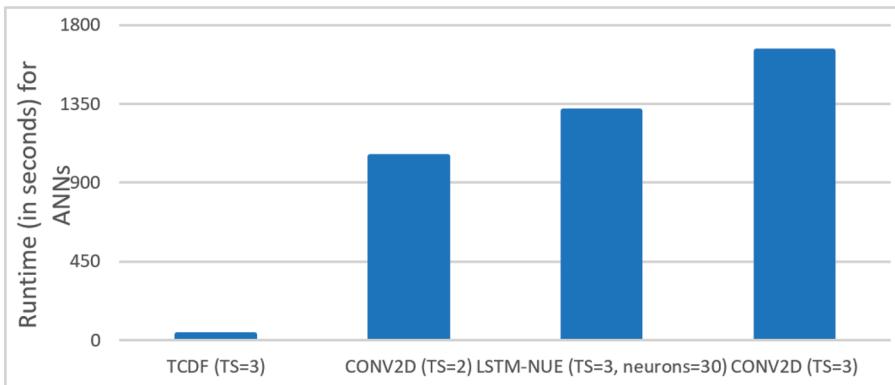


Figure 3.12: Runtime for the training of all ANNs, with 3 time series (TS = 3), L = 1500. An extra comparison showing the runtime of Conv2D with two time series as predictors (TS = 2) is shown (but all datasets contain 3 time series). “Neurons” = the number of hidden layer neurons.

	Far-Superficial		
	Sensitivity	Precision	F1
TRGC	1 ± 0.00	0.71 ± 0.29	0.79± 0.21
TCDF	0 ± 0.0	0 ± 0.0	0 ± 0
LSTM-NEU	1 ± 0.0	0.37 ± 0.13	0.53 ± 0.14
CONV2D * (TS=2)	0 ± 0.0	0 ± 0.0	0 ± 0.0
Conv2D (TS=3)	0 ± 0.0	0 ± 0.0	0 ± 0.0

Table 3.2: Scores of the ANNs in comparison with TRGC, using Ground Truth 2. Results are based upon datasets where all 3 time series (TS) were included as predictors, with one exception: results from Conv2D with two time series as predictors, indicated with *, were also included.

adjusted = 0.005), significant differences in Sensitivity and Precision between LSTM-NUE/TRGC versus all other methods were found ($p < 0.005$). No differences in Sensitivity between TRGC (Mdn = 1) and LSTM-NUE (Mdn = 1) were found ($p = 0.21$), while Precision was significantly higher for TRGC (Mdn = 0.67) than for LSTM-NUE (Mdn = 0.42), $p < 0.001$. It is not surprising that comparisons of any other ANN method than LSTM-NUE with TRGC were significant, given that these methods had a Sensitivity and Precision of zero. Even though dipole condition did not turn out to exhibit a significant effect on Sensitivity nor Precision in our data sets, this distinction remains theoretically important. We summarized the qualitative differences below.

	Far-Deep		
	Sensitivity	Precision	F1
TRGC	1 ± 0.00	0.70 ± 0.29	0.79 ± 0.21
TCDF	0 ± 0.0	0 ± 0.0	0 ± 0
LSTM-NEU	1 ± 0	0.43 ± 0.09	0.60 ± 0.09
CONV2D * (TS=2)	0 ± 0.0	0 ± 0.0	0 ± 0.0
Conv2D (TS=3)	0 ± 0.0	0 ± 0.0	0 ± 0.0

Table 3.3: Scores of the ANNs in comparison with TRGC, using Ground Truth 2. Results are based upon datasets where all 3 time series (TS) were included as predictors, with one exception: results from Conv2D with two time series as predictors, indicated with *, were also included.

Sensitivity and Precision were in both dipole conditions 0 while using TCDF and using two different configurations of Conv2D (once with two time series as predictors, once with three time series as predictors). In contrast, Precision was $M = 0.37 (\pm 0.13)$ for LSTM-NUE while TRGC obtained a precision of $M = 0.71 (\pm 0.29)$ in the Far-Superficial dipole condition. In the Far-Deep condition, performance of TRGC remained almost the same (Sensitivity $M = 1 \pm 0$, Precision $M = 0.70 \pm 0.29$) while it became slightly higher (in contrast to the Far-Superficial condition) for LSTM-NUE (Sensitivity $M=1\pm 0$, Precision $M = 0.43 \pm 0.09$). F1-scores were $M = 0.79 \pm 0.21$ and $M = 0.53 \pm 0.14$ for TRGC and LSTM-NUE, respectively, in the Far-Superficial condition and $M = 0.79 \pm 0.21$, $M = 0.60 \pm 0.09$ in the Far-Deep condition (while being zero for all the other ANNs).

3.5 Discussion

While considering Sensitivity and Precision, it was shown that, among the ANNs, LSTM-NUE yielded superior results in terms of Sensitivity, resulting in statistically significant differences with the other ANNs except for Conv2D with TS = 2. In terms of Precision, however, no significant differences among the ANNs were found while using Ground Truth 1. TRGC outperformed all ANNs in terms of Sensitivity but, statistically, no differences in Precision were found given that the main effect of the connectivity method was only marginally significant. The lack of a statistically significant effect of connectivity method on Precision, as well as the lack of an effect of dipole condition, and the lack of an interaction effect on both Sensitivity as well as Precision are quite counterintuitive. Indeed, given (1) the patterns observed across both Ground Truths and (2) the results from [1], which convincingly showed effects of different dipole conditions on

connectivity patterns as well as interaction effects of connectivity method and dipole condition, one could at least expect an effect of dipole condition. For instance, in [1], it was shown that with an SNR of 0.9 and in a Far–Superficial dipole condition, false positives (as related to Precision) were rather rare, while for other dipole conditions, the percentage of false positives increases (hence decreasing Precision). A related (solely qualitative) observation is the variability in the results of the ANNs (as became obvious through the standard deviations from the mean as depicted in Figures 3.7 and Figure 3.8) versus the stability of results produced by TRGC. In particular, ANNS seems to exhibit an increased variability in performance in the Far–Deep Condition (in contrast to the Far–Superficial condition), while almost no such variability is observed for TRGC. A possible culprit could be the initial randomization of the weights in ANNs, but how this instability could differ between architectures or between dipole conditions is unclear and deserves attention in future studies. One of the most important observations of Ground Truth 1 is the relatively poor Precision score of TRGC in the Far–Superficial condition, albeit that a difference with the Far–Deep dipole condition could not be statistically confirmed. More data may be needed to confirm the observed trends. The above-mentioned contrasting results are further discussed below, together with possible explanations with regard to the used connectivity methods.

Using Ground Truth 2, no differences in Sensitivity between TRGC and LSTM-NUE were found given that both methods returned almost always a Sensitivity of one, while Precision was significantly higher for TRGC than for LSTM-NUE. The other ANNs did not detect any connection. The good performance of TRGC regarding Precision is not surprising. In [1], it was already shown that TRGC outperformed Multivariate Granger Causality (MVG), especially when it comes to false positives (as reflected in a lower False Positive Rate), which is logical given that the introduction of time-reversal could indeed allow for a better distinction between correlated time series (due to linear mixtures of EEG signals) and true temporal precedence of one time series with regard to another. Although the idea of TRGC is relatively new (as it was first proposed in 2013, by [8]) in comparison to, for instance, bivariate GC and MVG, due to its appealing theoretical properties as well as its further validation by [7], it was quickly picked up in the field, given its relevance for, among others, EEG source connectivity. Recent developments include, for instance, variations in TRGC that allow for other than normal distributions [32]. In summary, it became clear that, among the ANNs, LSTM-NUE obtained better Sensitivity scores and (although only statistically confirmed using Ground Truth 2) better Precision scores. TRGC outperformed the ANNs in terms of Sensitivity, but in the case of Ground Truth 1, questions arose surrounding its Precision in the Far–Superficial dipole condition (although its Precision was significantly better in Ground Truth 2, without any indication of possible differences between

dipole conditions). While all connections were discovered, two false positives were detected relatively consistently, indicating that even with time-reversal there is, in certain circumstances, an overdetection of connections. The lack of performance of TCDF and Conv2D in Ground Truth 2 cannot be due to the location of the two fixed dipoles since they were located at the exact same location as in Ground Truth 1. Hence, we suspect that the moving nature of the sending dipole explains (at least partly) the lack of Sensitivity in TCDF and Conv2D. Taking the results from both Ground Truths together, both LSTM-NUE and TRGC are clearly more sensitive, but they both still tend towards overdetection.

With regard to the score strength rankings, not much can be said about TCDF given that the mean attention scores were significant only for two time series in the Far-Superficial dipole condition, from which one was a falsely detected connectivity (i.e., a false positive). In contrast to TCDF, with LSTM-NUE, for two out of three targets, correct column-wise rankings were obtained for Ground Truth 1. For Conv2D (with TS = 2), correct rankings for predictor pair were found in terms of R2-scores, also for two out of three targets. When looking closer to the contributions of individual time series, it was found that predicting, for instance, X1, with itself and another time series works better than predicting it without the past of X1, which is logical. The fact that adding more predictors (i.e., Conv2D with TS = 3) did not work out is obviously the most problematic aspect of Conv2D. Once a third predictor was added, performance dropped substantially, and it was hypothesized that this could be due to the fact that it was convolving rather uncorrelated or only slightly correlated time series together confuses the two-dimensional network to the extent that no proper prediction can be made. The fact that channels are not kept separate such as in a depthwise-separable architecture, may play an important role in this aspect. Finally, with regard to runtimes (time needed to train a model), LSTM-NUE was together with Conv2D, TS = 3 the most time-consuming method, which calls for a trade-off between accuracy and Time Complexity. It is especially the non-uniform embedding strategy (NUE) that is responsible for the high Time Complexity. However, in [15], it was shown that the current LSTM-model could also produce reasonable results without implementation of the NUE strategy, thereby lowering its Time Complexity drastically.

Moreover, in [15], it was shown that LSTM-NUE could cope with different types of ground truths (linear, non-linear and non-linear with varying length lags), as confirmed in our work. Contrary to [15], we, in addition, had Ground Truth 2 with a moving dipole (i.e., the “Sender”) which worked relatively well for LSTM-NUE. Hence, the latter can cope not only with time-varying parameters but also, to some extent, with changing dipole locations. Both TCDF and Conv2D cope far less well with a moving sender, probably (or at least partly) because of

the occurrence of both closeness and deepness in the same setting, which has an impact on how signals are transformed by source reconstruction.

TCDF and Conv2D are, in contrast to LSTM-NUE, not a part of the family of Recurrent Neural Networks and therefore do not contain feedback loops. The LSTM is particularly known for its excellent memory properties by virtue of its gates that help to remember versus forget certain time samples. In general, the better memory properties of an LSTM in combination with the NUE approach probably play an important role in dealing with variations over time. An LSTM may also be better in looking through (uncorrelated) noise components because it remembers formerly seen time samples better and, subsequently, should be better in detecting (even weak) patterns over time, also when occluded by noise. This, in turn, may make it easier to deal with more challenging dipole locations or with heavier data transformations. However, this same property could also make an LSTM more sensitive to correlated noise from source mixing. TCDF, on the other hand, has the advantage of a very low Time Complexity, at least partly due to its sparsity in interconnection weights (given its depthwise-separable architecture), but it seems less able to distinguish correlation from causation. This may be due to the lack of feedback loops, an “active” memory feature that makes it difficult to distinguish true patterns from noise over longer time intervals. In this study, TCDF was tuned as such that not too many false positives were detected (given its problem of distinguishing correlation from causation), and this more “conservative” configuration may have led to its low Sensitivity. Overall, we can conclude that, among the ANN models, LSTM-NUE performed best in terms of Sensitivity and Precision regardless of which ground truth was used even though no shuffling or time-reversal was used for connectivity assessment. The contrasting results of TRGC in terms of Precision between dipole conditions in Ground Truths 1 and 2 are puzzling and clearly show an “oversensitivity” of TRGC under certain circumstances. Still, TRGC and LSTM-NUE yielded acceptable-to-good results, albeit both suffer from over-detection. An interesting new finding is the fact that an LSTM is, to some extent, able to provide an answer to the question of whether connectivity between sources is present or absent, at least for source-reconstructed, simulated EEG data. The fact that too many faulty connections were detected (especially in Ground Truth 2) calls for improvements. One possibility is to use LSTM-NUE as part of a masking approach, on top of which another learner is stacked. This masking approach has already led to many advantages in source localization [25], and it may also facilitate connectivity detection with ANNs, especially when overly sensitive to it. In this sense, other ANNs, even with a lower Time Complexity than that of LSTM-NUE, could possibly also be considered as potentially directed connectivity estimators.

An obvious future step is testing whether ANNs can also be applied to real EEG

data, albeit that several possible caveats should be taken into account. First and foremost, as shown by [1], under low noise conditions, dipole conditions may matter less, but differences between dipole conditions could become more obvious (i.e., more disturbing) under higher noise levels. Even long-established connectivity methods suffer from this. Since controlling noise levels is hard, reasonably one could opt for EEG-data for which (1) the contributing brain areas are rather superficially located, (2) the connectivity patterns are relatively well known and preferably supported by both high-density EEG and fMRI-data so that a performance evaluation becomes feasible since no ground truth is available for real EEG-data. Testing ANNs and contrasting them with TRGC/other established methods using vision-related or motor-related EEG-datasets makes thus more sense than testing them with data with relatively unknown connectivity patterns. Regions of Interest (ROIs) can be defined based upon previous knowledge about involved brain areas. As for source localization, a reasonable choice is eLORETA. Data-driven approaches (as opposed to ROI-selection), e.g., data-driven clustering [33], seem only reasonable in a later stage when the value of the used ANN is proven on real EEG data.

3.6 Conclusions

Some types of neural networks, in particular LSTMs, may be considered for estimating the directed connectivity of reconstructed EEG Sources. However, no method is flawless, and we showed that even an established method such as TRGC can generate faulty estimates. This calls for further developments. There is much potential for a hybrid approach, in which a neural network could be used as a preprocessing step to chart the interesting directed connectivity patterns, after which a conventional method is applied for estimating them.

3.7 Acknowledgement

3.7.1 Author Contributions

Conceptualization, M.M.V.H., A.F. and I.V.; methodology, I.V. and A.F.; formal analysis, I.V.; resources (scripts), A.F. and I.V.; original draft preparation, I.V.; writing, review and editing, A.F and M.M.V.H.; visualization, I.V.; supervision, A.F. and M.M.V.H.; project administration, A.F. All authors have read and agreed to the published version of the manuscript.

3.7.2 Funding

I.V. performed this work as part of her Master in Artificial Intelligence thesis at KU Leuven. A.F. is supported by a grant from the Belgian Fund for

Scientific Research—Flanders (FWO 1157019N). M.M.V.H. is supported by research grants received from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857375, the special research fund of the KU Leuven (C24/18/098), the Belgian Fund for Scientific Research—Flanders (G0A4118N, G0A4321N, G0C1522N), and the Hercules Foundation (AKUL 043).

3.7.3 Conflicts of Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Bibliography

- [1] A. Anzolin, P. Presti, F. Van De Steen, L. Astolfi, S. Haufe, and D. Marinazzo, “Quantifying the effect of demixing approaches on directed connectivity estimated between reconstructed eeg sources”, *Brain topography*, vol. 32, no. 4, pp. 655–674, 2019.
- [2] L. BACCALÁ, “A y sameshima, k., 2001. partial directed coherence: A new concept in neural structure determination”, *Biological cybernetics*, pp. 463–74,
- [3] K. J. Friston, L. Harrison, and W. Penny, “Dynamic causal modelling”, *Neuroimage*, vol. 19, no. 4, pp. 1273–1302, 2003.
- [4] C. Bchel and K. Friston, “Modulation of connectivity in visual pathways by attention: Cortical interactions evaluated with structural equation modeling and fmri”, *Cerebral Cortex*, vol. 7, no. 8, p. 76878, 1997.
- [5] C. Bernasconi and P. KoÈnig, “On the directionality of cortical interactions studied by structural analysis of electrophysiological recordings”, *Biological cybernetics*, vol. 81, no. 3, pp. 199–210, 1999.
- [6] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, “Causal effect inference with deep latent-variable models”, *Advances in neural information processing systems*, vol. 30, 2017.
- [7] I. Winkler, D. Panknin, D. Bartz, K.-R. Muller, and S. Haufe, “Validity of time reversal for testing granger causality”, *IEEE Transactions on Signal Processing*, vol. 64, no. 11, pp. 2746–2760, 2016.
- [8] S. Haufe, V. V. Nikulin, K.-R. Müller, and G. Nolte, “A critical assessment of connectivity measures for eeg data: A simulation study”, *Neuroimage*, vol. 64, pp. 120–133, 2013.
- [9] H. Maier and G. Dandy, “Artificial neural networks: A flexible approach to modelling”, 2004.
- [10] S. Leijnen and F. v. Veen, “The neural network zoo”, in *Proceedings*, MDPI, vol. 47, 2020, p. 9.

- [11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling”, *arXiv preprint arXiv:1412.3555*, 2014.
- [12] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, “Learning precise timing with lstm recurrent networks”, *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.
- [13] Y. He and J. Zhao, “Temporal convolutional networks for anomaly detection in time series”, in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1213, 2019, p. 042050.
- [14] M. Nauta, D. Bucur, and C. Seifert, “Causal discovery with attention-based convolutional neural networks”, *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 312–340, 2019.
- [15] Y. Wang, K. Lin, Y. Qi, Q. Lian, S. Feng, Z. Wu, and G. Pan, “Estimating brain connectivity with varying-length time lags using a recurrent neural network”, *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 9, pp. 1953–1963, 2018.
- [16] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, “Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting”, *Electronics*, vol. 8, no. 8, p. 876, 2019.
- [17] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling”, *arXiv preprint arXiv:1803.01271*, 2018.
- [18] Q. Fu, C. Hu, J. Peng, and S. Yue, “Shaping the collision selectivity in a looming sensitive neuron model with parallel on and off pathways and spike frequency adaptation”, *Neural Networks*, vol. 106, pp. 127–143, 2018.
- [19] M. Lai, M. Demuru, A. Hillebrand, and M. Fraschini, “A comparison between scalp-and source-reconstructed eeg networks”, *Scientific reports*, vol. 8, no. 1, p. 12269, 2018.
- [20] A. Omidvarnia, M. Mesbah, J. M. O’Toole, P. Colditz, and B. Boashash, “Analysis of the time-varying cortical neural connectivity in the newborn eeg: A time-frequency approach”, in *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, IEEE, 2011, pp. 179–182.
- [21] M. Fahimi Hnazaee, E. Khachatryan, S. Chehrazad, A. Kotarcic, M. De Letter, and M. M. Van Hulle, “Overlapping connectivity patterns during semantic processing of abstract and concrete words revealed with multivariate granger causality analysis”, *Scientific reports*, vol. 10, no. 1, p. 2803, 2020.
- [22] M. Winterhalder, B. Schelter, W. Hesse, K. Schwab, L. Leistritz, D. Klan, R. Bauer, J. Timmer, and H. Witte, “Comparison of linear signal processing techniques to infer directed interactions in multivariate neural systems”, *Signal processing*, vol. 85, no. 11, pp. 2137–2160, 2005.

- [23] Y. Huang, L. C. Parra, and S. Haufe, “The new york head (icbm-ny)–description of the provided data”, 2017.
- [24] R. D. Pascual-Marqui, “Discrete, 3d distributed, linear imaging methods of electric neuronal activity. part 1: Exact, zero error localization”, *arXiv preprint arXiv:0710.3341*, 2007.
- [25] A. Faes, A. de Borman, and M. Van Hulle, “Source space reduction for eloreta”, *Journal of Neural Engineering*, vol. 18, no. 6, p. 066014, 2021.
- [26] R. D. Pascual-Marqui, D. Lehmann, M. Koukkou, K. Kochi, P. Anderer, B. Saletu, H. Tanaka, K. Hirata, E. R. John, L. Prichep, *et al.*, “Assessing interactions in the brain with exact low-resolution electromagnetic tomography”, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 369, no. 1952, pp. 3768–3784, 2011.
- [27] Nauta. “Temporal causal discovery framework”. (2021), [Online]. Available: <https://github.com/M-Nauta/TCDF> (visited on 09/22/2021).
- [28] Feng. “Rnn-gc”. (2021), [Online]. Available: <https://github.com/shaozhefeng/RNN-GC> (visited on 07/15/2021).
- [29] J. Wales. “Time series forecasting with 2d convolutions.” (2021), [Online]. Available: <https://towardsdatascience.com/time-series-forecasting-with-2d-convolutions-4f1a0f33dff6> (visited on 07/15/2021).
- [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python”, *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [31] M. Gerstung, E. Papaemmanuil, and P. J. Campbell, “Subclonal variant calling with multiple samples and prior knowledge”, *Bioinformatics*, vol. 30, no. 9, pp. 1198–1204, 2014.
- [32] M. Chvostekova, J. Jakubik, and A. Krakovska, “Granger causality on forward and reversed time series”, *Entropy*, vol. 23, no. 4, p. 409, 2021.
- [33] S. H. Wang, M. Lobier, F. Siebenhühner, T. Puoliväli, S. Palva, and J. M. Palva, “Hyperedge bundling: A practical solution to spurious interactions in meg/eeg source connectivity analyses”, *NeuroImage*, vol. 173, pp. 610–622, 2018.

Chapter 4

Single Finger Trajectory Prediction From Intracranial Brain Activity Using Block-Term Tensor Regression With Fast and Automatic Component Extraction

AXEL FAES, FLAVIO CAMARRONE AND MARC M. VAN HULLE

IEEE Transactions on Neural Networks and Learning Systems
(2022).

4.1 Abstract

Multiway- or tensor-based decoding techniques for Brain Computer Interfaces (BCI) are believed to better account for the multilinear structure of brain signals than conventional vector- or matrix-based ones. However, despite their outlook on significant performance gains, the used parameter optimization approach is often too computationally demanding so that conventional techniques are still

preferred. We propose two novel tensor factorizations which we integrate into our Block-Term Tensor Regression (BTTR) algorithm and further introduce a marginalization procedure that guarantees robust predictions while reducing the risk of overfitting (generalized regression). BTTR accounts for the underlying (hidden) data structure in a fully automatic and computationally-efficient manner, leading to a significant performance gain over conventional vector- or matrix-based techniques in a challenging real-world application. As a challenging real-world application, we apply BTTR to accurately predict single finger movement trajectories from intracranial recordings in human subjects. We compare the obtained performance with that of the state-of-the-art.

4.2 Introduction

Brain Computer Interfaces (BCIs) decode the user’s intent from his/her brain activity directly, bypassing the need for muscular control. Detection requires computational processing to separate (“decode”) activity related to the intended action from background noise after which the result is sent to an actuator (see review [1]). The ability to control a robotic limb or regain control over a paralyzed limb with “motor-BCIs” has promoted the latter as a solution for patients deprived from voluntary movement, but that are otherwise fully conscious, due to spinal cord injury, brain stem stroke or a degenerative disorder such as amyotrophic lateral sclerosis (ALS). Extraordinary results have been achieved with functional electrical stimulation (FES) of hand muscles [2], [3] and prosthetic hands and arms, exoskeletons or other effectors [4]–[7].

In the last decade, BCI researchers are paying increased attention to electrocorticography (ECoG) where brain activity is recorded via grids or strips of electrodes placed intracranially on the cortical surface. Compared to multi-electrode and deep brain implants, ECoG offers larger coverages of cortical surface combined with more long-term signal stability [8], [9] and, compared to electroencephalography (EEG), a higher spatial resolution [10] and a larger spectral bandwidth and signal amplitude [11]. Several studies [12]–[18] reported that ECoG signals recorded from the primary motor cortex can promote successful decoding of continuous arm trajectories. Nevertheless, accurate decoding of more subtle muscular activity such as single finger extension/flexion is considered the next challenge.

Currently, ECoG-based decoding approaches for finger movements have been proposed based mainly on conventional sparse [19], gaussian [20], linear regression [21], [22], Convolutional Neural Networks (CNN), Random Forests (RF) and deep learning networks (proposed in [23]). However, despite the encouraging results, we advocate that finger trajectory decoding could benefit from recent developments in multilinear algebra (multiway decoding) as ECoG

is in essence structured in space, time, and frequency domain. For instance, in relation to muscular actions, a distinctive spatio-temporal-spectral activity pattern can be observed in the motor cortex: a decrease in activity in both the mu and beta bands following movement onset termed event-related desynchronisation (ERD) (Graimann et al., 2011), and an increase in beta band activity following movement termination termed event-related synchronisation (ERS) [24], whereby the spatial distribution of both depends on the considered movement task. When relying on a conventional vector- or matrix-based regression model, the original multimodal structure is largely ignored as the data is concatenated into vectors and matrices (aka unfolding), which could cause the model to underperform and hamper its interpretation [25]. Furthermore, the size of the vectorized/matrixized data calls for a model with a large number of tunable parameters rendering it susceptible to noise and redundancy [26]. Modern deep learning methods have also been proposed, such as those based on Riemannian features [27].

On the other hand, multiway models preserve the multilinear structure of the data and support the discovery of potentially hidden multilinear components [28]. For this reason, they have attracted interest from researchers in neuroimaging, image and video completion, numerical analysis, data mining, etc. The two most popular multiway frameworks are Tucker (TKD) and CANDECOMP/PARAFAC (or CPD) decompositions (see [28], [29] for review) and both aim to determine the low-dimensional space where important information is residing. While the former seeks a low multilinear rank (MTR) approximation of the tensor data, the latter approximates the tensor data as a sum of rank-one tensors. Note that the concept of rank is associated with the "information content" of the tensor: the lower the rank, the lower the information content. Clearly, identifying the true rank of a tensor is the key to accurate multiway analysis. De Lathauwer introduced a tensor decomposition algorithm, called Block Term Decomposition (BTD) [30]–[32], which can be seen as a unified version of TKD and CPD. More specifically, in [32] two BTD models were proposed to approximate N^{th} -order tensor data as a sum of K-blocks called rank- (L_1^k, \dots, L_N^k) BTD and rank- (L_1, \dots, L_N) BTD with $k=1, \dots, K$. In the former model blocks can have different MTR, whence in the latter a unique MTR is chosen for all blocks. If there is only one block, then BTD reduces to TKD. In contrast, if all blocks have multilinear rank MTR = $(1, 1, \dots, 1)$, then it reduces to CPD. This approach provides great flexibility and opens new possibilities for multiway data analysis, above all when rank- (L_1^k, \dots, L_N^k) BTD is adopted. For instance, a recent EEG study [33] showed that, while CPD failed to model epileptic seizures, rank- (L_1, \dots, L_N) BTD correctly extracted the ictal sources that matched clinical assessment.

The abovementioned multiway frameworks have been adapted for regression

analysis to model the relationship between arm trajectory and ECoG signals recorded from monkeys [34], [35] and recently between exoskeleton-based arm trajectory, arm- and wrist rotations and ECoG signals recorded from a tetraplegic patient with a spinal cord injury [18]. The decoder in the latter case relies on a variant of multiway partial least squares regression (NPLS), but is not capable of achieving the kind of accuracy needed to decode fine limb movements. Possible reasons for this underperformance are the limited fitness ability, the high computational complexity and the slow convergence of NPLS when handling higher-order data [34]. On the other hand, Zhao et al. [34] developed a powerful generalized framework, called Higher-Order Partial Least Squares (HOPLS), based on $(1, L_2, \dots, L_N)$ -rank BTD (i.e. all blocks have the same MTR), that provides enhanced predictability with optimal balance between fitness and model complexity. As a result, HOPLS was able to outperform conventional PLS.

However, the success of HOPLS and other tensor-based regression methods largely depends on the selection of appropriate model parameters, e.g., the number of latent variables, a challenging issue and the subject of ongoing research. Evidently, cross-validation could be used to identify the (near) optimal parameter combination by assessing all (or as many as desired) parameter combinations, but the ensuing computational effort could be too prohibitive to be practical, above all in time-critical applications such as BCI. In addition, as current multiway regression models use fixed MTRs, they do not fully exploit the potential of rank- (L_1^k, \dots, L_N^k) BTD, above all in cases where the true rank changes across blocks.

In order to accurately decode single finger movements from ECoG signals, a challenging real-world application, we propose a fast and flexible multiway model based on (L_1^k, \dots, L_N^k) BTD (namely BTTR) with automatic parameter selection. In this way, we tackle the computationally intensive parameter estimation assumed by state-of-the-art tensor-based methods as well as cases where the optimal MTR varies across blocks. Further we introduce a marginalization procedure to obtain robust predictions while reducing the risk of overfitting (generalized regression). Our motivation for both developments is to promote tensor-based techniques in particular for BCI purposes. In the next sections, we introduce a new TKD method for automatic rank selection and latent component extraction given two more or less correlated multiway variables, called Automatic Component Extraction (ACE), and its improved version augmented with automatic latent component selection, called Automatic Correlated Component Selection (ACCoS). The ACE or ACCoS models are embedded in the BTTR process to sequentially extract the components that maximize the correlation between multiway response and prediction. We then applied BTTR on ECoG data taken from BCI competition IV [36]. As a result, our model yields a higher

accuracy compared to HOPLS, linear regression [22], Random Forests (RF), Convolutional Neural Networks (CNN), and Long Short-Term Memory Network (LSTM) [23]. In this way, we wish to promote the use of multiway modelling for real-world BCI applications where accurate and fast decoding of brain activity is required as in ECoG-based neuroprostheses.

4.3 Previous work on automatic multilinear tensor rank selection

Large tensor data usually contains intrinsically low-dimensional information extractable with tensor frameworks such as TKD, a form of higher-order principal component analysis that is adopted in a broad range of applications [37]. Given a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, TKD decomposes $\underline{\mathbf{X}}$ into a smaller tensor (core tensor) and N factor matrices: $\underline{\mathbf{X}} \approx \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \dots \times_N \mathbf{A}^{(n)}$ with $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_n}$ the core tensor and $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ the factor matrix associated with the n th mode. The obtained decomposition is often not unique so that constraints such as orthogonality, sparsity or nonnegativity are generally imposed on the factor matrices and/or core tensor [38]–[40]. For instance, Higher-Order Singular Value Decomposition (HOSVD) [38] and Higher-Order Orthogonal Iteration (HOOI) [41] were proposed as variants of TKD with orthogonality constraints on the factor matrix.

The multilinear rank (MTR) is the tuple $R_1 \times \dots \times R_n$ that defines the size of the factor matrices and, hence, the core tensor. The choice of MTR is generally considered critical: a large MTR might lead to an approximation containing uninteresting information; a small MTR might yield a low compression ratio incapable of fully representing complex tensor data. Recently, researchers have been focusing on determining the MTR, and thus model complexity, in an automatic way. For instance, Mørup and Hansen [42] proposed a method based on Bayesian learning for sparse Tucker decomposition. In this method, called ARD Tucker, the core tensor and matrices are alternately and iteratively updated, while the number of components in each mode are determined using automatic relevance determination (ARD). Yokota et al. [43] proposed pruning sparse Tucker decomposition (PSTD) of which the objective is to minimize the L1-norm of the core tensor under conditions of error bound and orthogonality constraints of individual basis matrices. At each iteration, the factor matrices and the sparsity of the core tensor are updated, and unnecessary dimensions removed according to the entries of the latter. More recently, Yokota et al [44] published a Tucker-rank estimation approach using robust minimum description length (MDL): the rank R_n is estimated by applying the MDL criterion on the distribution of the eigenvalues extracted from the n th mode unfolded core tensor via HOSVD. This method is referred to as SCORE. Shi et al. [45] proposed

a multilinear Tensor Rank Estimation based on L1-regularized orthogonal CP decomposition (TREL1). Using a block coordinate descent approach, the CP components and corresponding weight vectors are iteratively updated. Finally, TREL1 automatically determines the MTR by pruning the zero entries of the weight vector.

4.4 Methods

One of the main limitations of HOPLS, the current state-of-the-art multiway regression models, is the assumed prior knowledge of the model parameters, but this information is commonly not available for a given real-word application. Therefore, the computationally expensive cross-validation is often used to identify these parameters. In addition, HOPLS can be seen as a BTD-like approach where each block has the same $(1, L_2, \dots, L_N)$ -rank which, in turn, limits flexibility in data modelling. To tackle this, we first define a novel tensor decomposition model with automatic MTR determination with the intent to extract the components that are maximally correlated between two variables. Then, we include the proposed model into a deflation-based method for multiway regression, namely BTTR (Block Term Tensor Regression), for high flexibility and interpretability. An overview of the mathematical notation used can be found in Table 4.1.

Using BTTR first requires formatting the data into a tensor format. BTTR will work iteratively. Each iteration either ACE or ACCoS is used for automatic parameter estimation. This is followed by deflation before the next iteration starts.

4.4.1 Component extraction and selection

Given an N-way variable $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and a vectoral variable $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$, we aim to automatically extract the latent components \mathbf{t} and $P^{(n)}_{(n=2)}$, associated with the n-th mode of $\underline{\mathbf{X}}$ and maximally correlated with \mathbf{y} , while $\|\underline{\mathbf{X}}\| - \|\underline{\mathbf{G}}; \mathbf{t}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}\|_F$ is minimized. To this end, we first introduce a novel model for tensor decomposition, characterized by robust component extraction. Then, we extend the proposed model with correlated component selection. An overview of the proposed models is reported in Figure 1.

1) Modified PSTD for component extraction

Our new method for determining MTR uses sparsity constraints on the core tensor to prune irrelevant components. This idea is inherited from the PTSD model which has recently proved to be a valid tool for MTR selection. We will refer to the proposed model as modified PSTD (mPSTD). The mPSTD model

Table 4.1: Mathematical notation

Notation	Description
$\underline{\mathbf{T}}, \mathbf{M}, \mathbf{v}, S$	tensor, matrix, vector, scalar (respectively)
\mathbf{M}^T	transpose of matrix
\times_n	mode-n product between tensor and matrix
\otimes	Kronecker product
\circ	outer product
$\ \cdot\ _F$	Frobenius norm
$\mathbf{T}_{(n)}$	mode-n unfolding of tensor $\underline{\mathbf{T}}$
$\underline{\mathbf{C}}^{(T)}$	core tensor associated to tensor $\underline{\mathbf{T}}$
$\mathbf{M}^{(n)}$	mode-n factor matrix
\mathbf{M}_{ind}	(sub-)matrix including the column(s) indicated in ind
$\mathbf{M}_{\setminus ind}$	(sub-)matrix excluding the column(s) indicated in ind
$\llbracket \underline{\mathbf{C}}; \mathbf{M}^{(1)}, \dots, \mathbf{M}^{(N)} \rrbracket$	full multilinear product $\underline{\mathbf{C}} \times_1 \mathbf{M}^{(1)} \times_2 \dots \times_N \mathbf{M}^{(N)}$
$\langle \underline{\mathbf{T}}, \underline{\mathbf{E}} \rangle_{\{n,n\}}$	mode-n cross-covariance tensor

is first initialized with HOSVD. Then, iteratively, a soft-thresholding rule based on parameter λ , alternated with a threshold τ –adopted from standard PSTD–, are applied to enhance model sparsity and to prune irrelevant components, respectively. Note that in [43] $\text{SNR} \in [1, 50]$ is used to derive, via a line search, the optimal degree of sparsity λ of the core tensor (see [43] for a derivation of λ). At each iteration, the core tensor $\underline{\mathbf{G}}$ is updated using the soft-thresholding rule as $\underline{\mathbf{G}} = sgn(\underline{\mathbf{G}}) \times max\{|\underline{\mathbf{G}}| - \lambda, 0\}$, while the threshold $\tau \in [0, 100]$ is used to reject unnecessary components from the n-mode $S^{(n)} = \{r | 100(1 - \frac{\sum_i \mathbf{G}_{(n)(r,i)}}{\sum_{t,i} \mathbf{G}_{(n)(t,i)}}) \geq \tau\}$, $\mathbf{P}^{(n)} = \mathbf{P}^{(n)}(:, S^{(n)})$ and $\mathbf{G}^{(n)} = \mathbf{G}^{(n)}(S^{(n)}, :)$. The mPTSD is summarized in Algorithm 4.1.

2) Proposed ACE for automatic component extraction

Importantly, in standard and modified PSTD, parameters such as noise level of the data (SNR) and the threshold for component rejection (τ) are assumed to be

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$, τ , SNR
Output: $\underline{\mathbf{G}} \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$, $\{\mathbf{P}^{(n)}\}_{n=2}^N$

Initialisation :

- 1: $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times \dots \times I_N}$
- 2: **Initialisation of** $\{\mathbf{P}^{(n)}\}_{n=2}^N$ and $\underline{\mathbf{G}}$ using HOOI on $\underline{\mathbf{C}}$
- LOOP Process*
- 3: **repeat**
- 4: **update** $\underline{\mathbf{G}}$ using SNR
- 5: **prune** $\{\mathbf{P}^{(n)}\}_{n=2}^N$ and $\underline{\mathbf{G}}$ using τ
- 6: **until** convergence is reached
- 7: **return** $\underline{\mathbf{G}}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Figure 4.1: mPSTD

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$
Output: $\underline{\mathbf{G}}^{(X)} \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$, $\mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

- 1: $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times \dots \times I_N}$
- 2: **Initialisation of** $\tau = 90, \dots, 100$; $SNR = 1, \dots, 50$
- 3: **for** SNR_i in SNR **do**
- 4: **for** τ_j in τ **do**
- 5: $\underline{\mathbf{G}}, \{\mathbf{P}^{(n)}\}_{n=2}^N = mPSTD(\underline{\mathbf{X}}, \mathbf{y}, SNR_i, \tau_j)$
- 6: calculate BIC value corresponding to SNR_i and τ_j using Eq 4.1
- 7: **end for**
- 8: **select** $\tau^* = \operatorname{argmin}_\tau \text{BIC}(\tau)$
- 9: calculate BIC value corresponding to SNR_i and τ^* using Eq 4.1
- 10: **end for**
- 11: **select** $SNR^* = \operatorname{argmin}_{SNR} \text{BIC}(SNR, \tau^*)$
- 12: $\underline{\mathbf{G}}, \{\mathbf{P}^{(n)}\}_{n=2}^N = mPSTD(\underline{\mathbf{X}}, \mathbf{y}, SNR^*, \tau^*)$
- 13: $\mathbf{t} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(N)T})_{(1)} \operatorname{vec}(\underline{\mathbf{G}})$
- 14: $\mathbf{t} = \mathbf{t} / \|\mathbf{t}\|_F$
- 15: $\underline{\mathbf{G}}^{(X)} = [\underline{\mathbf{X}}; \mathbf{t}^T, \mathbf{P}^{(2)T}, \dots, \mathbf{P}^{(N)T}]$
- 16: **return** $\underline{\mathbf{G}}^{(X)}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Figure 4.2: ACE

known. However, in reality, this is not always the case and can have an important impact on model performance. Let us now define the mode-1 cross-product between predictor and response variables as $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times \dots \times I_N}$ and its decomposition as $\underline{\mathbf{C}} \approx [\underline{\mathbf{G}}^{(c)}; \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}]$. We provide the model with automatic SNR and τ selection based on Bayesian Information Criterion (BIC) defined here as:

$$BIC(\tau, \text{SNR} | \text{SNR}, \tau^*) =$$

$$\log\left(\frac{\|\underline{\mathbf{C}}\| - \|\underline{\mathbf{G}}^{(c)}; \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}\|_F}{s}\right) + \frac{\log(s)}{s} DF \quad (4.1)$$

where $\underline{\mathbf{G}}^{(c)}$ and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are the sparse core and factor matrices obtained with mPSTD using specific τ and SNR values, s the number of entries in $\underline{\mathbf{G}}$, and DF the degree of freedom calculated as the number of non-zero elements in $\underline{\mathbf{G}}^{(c)}$, as suggested in [46]. BIC is well known for its consistency in selecting the true model [47] as it is based on a trade-off between model fit and -complexity. A lower BIC value indicates a better candidate. For each SNR value, the associated optimal τ is computed as $\tau^* = \operatorname{argmin}_\tau BIC(\tau, \text{SNR})$. Then, the optimal SNR is determined as $\text{SNR}^* = \operatorname{argmin}_{\text{SNR}} BIC(\text{SNR}, \tau^*)$. Once $\underline{\mathbf{G}}^{(c)}$ and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are computed, the score vector \mathbf{t} is first calculated as

$$\mathbf{t} = (\underline{\mathbf{C}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(n)T})_{(1)} \operatorname{vec}(\underline{\mathbf{G}}^{(c)}),$$

and then normalized. We refer to this fully automatic component extraction as ACE, summarized in Algorithm 4.2.



Figure 4.3: Overview of the proposed models for component extraction and selection.

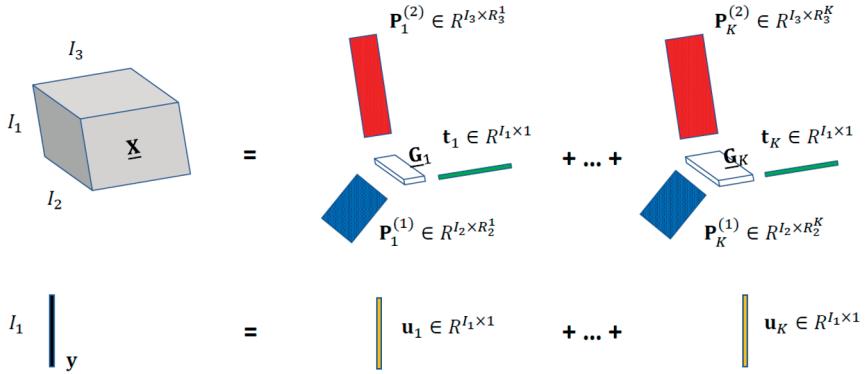


Figure 4.4: Scheme of BTTR algorithm with 3rd-order predictor variable $\underline{\mathbf{X}}$ and 1st-order response variable \mathbf{y} . Note that each block is computed using either ACE or ACCoS further referred to as ACE-BTTR and ACCoS-BTTR.

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$
Output: $\underline{\mathbf{G}}^{(X)} \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$, \mathbf{t} , $\{\mathbf{P}^{(n)}\}_{n=2}^N$
1: $\underline{\mathbf{G}}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N = ACE(\underline{\mathbf{X}}, \mathbf{y})$
2: select correlated components using Eq 4.2 and Eq 4.3
3: $\mathbf{t} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(N)T})_{(1)} vec(\underline{\mathbf{G}})$
4: $\mathbf{t} = \mathbf{t} / \|\mathbf{t}\|_F$
5: $\underline{\mathbf{G}}^{(X)} = [\underline{\mathbf{X}}; \mathbf{t}^T, \mathbf{P}^{(2)T}, \dots, \mathbf{P}^{(N)T}]$
6: return $\underline{\mathbf{G}}^{(X)}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Figure 4.5: ACCoS

3) Automatic correlated component selection (ACCoS)

Once the core tensor $\underline{\mathbf{G}}^{(c)}$ and factors $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are extracted via ACE, we select only the relevant components in a fully automatic manner as well. The full process consists of two steps: scoring (Step 1) and grouping (Step 2). In Step 1, for each n-mode factor, the single rth component is scored using the R-squared test between $\mathbf{x}_r^{(n)}$ and \mathbf{y} where

$$\mathbf{x}_r^{(n)} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_n \mathbf{P}_{\setminus r}^{(n)T} \times_{n+1} \dots \times_N \mathbf{P}^{(N)T})_{(1)} \\ vec(\underline{\mathbf{G}}_{(n) \setminus r}^{(c)}) \quad (4.2)$$

with $\underline{\mathbf{G}}_{(n)\setminus r}^{(c)} \in \mathbb{R}^{(R_n-1) \times (R_1 \times \dots \times R_N)}$ the n-mode matricization of the core tensor $\underline{\mathbf{G}}^{(c)}$ of which row r is removed. A lower score indicates a higher relevance of the associated (removed) component in the overall correlation between $\underline{\mathbf{X}}$ and \mathbf{y} .

In Step 2, we iteratively group the most relevant components. In order to select the smallest number of components that maximizes correlation, we start with the component with the lowest score in Step 1 and, then, iteratively add the one with the next lowest score. At each iteration, a new score is calculated using the R-squared test between $\mathbf{x}_{\text{ind}}^{(n)}$ and \mathbf{y} where

$$\mathbf{x}_{\text{ind}}^{(n)} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_n \mathbf{P}_{\text{ind}}^{(n)T} \times_{n+1} \dots \times_N \mathbf{P}^{(N)T})_{(1)} \text{vec}(\underline{\mathbf{G}}_{(n)\text{ind}}^{(c)}) \quad (4.3)$$

with ind the index of the first D components with lowest score and $\underline{\mathbf{G}}_{(n)\text{ind}}^{(c)} \in \mathbb{R}^{D \times (R_n \times \dots \times R_N)}$ the mode-n core tensor $\underline{\mathbf{G}}^{(c)}$ in which D rows are selected. The selection process stops when the score starts to decrease. Finally, like ACE, given the new core tensor $\underline{\mathbf{G}}^{(c)}$ and factor matrices $\{\mathbf{P}^{(n)}\}_{n=2}^N$ with selected components, the score vector \mathbf{t} is first computed as

$$\mathbf{t} = (\underline{\mathbf{C}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(n)T})_{(1)} \text{vec}(\underline{\mathbf{G}}^{(c)})$$

and then normalized. The complete framework is summarized in Algorithm 4.5 and further referred to as ACCoS.

4.4.2 Block-Term Tensor Regression (BTTR)

We propose a novel Block-Term Regression (BTTR) model based on (L_1^k, \dots, L_N^k) BTD with automatic MTR determination. More specifically, BTTR is a deflation-based method in which the maximally correlated representations of $\underline{\mathbf{X}}$ and \mathbf{y} are extracted via ACE/ACCoS at each iteration. Therefore, BTTR inherits the advantages of the proposed ACE/ACCoS and does not require one to set the model parameters manually. This provides BTTR with an additional important property: the ability to model complex data in which the optimal MTR is not necessarily stable across sequential decompositions. A scheme of BTTR is shown in Figure 2 whereas the full process is shown in Algorithm 4.6. In the rest of the paper, when required, we will use the terms ACE-BTTR and ACCoS-BTTR to refer to ACE and ACCoS in BTTR, respectively.

```

Input:  $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ ,  $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$ ,  $K$ 
Output:  $\{\mathbf{P}_k^{(n)}\}, \{\mathbf{t}_k\}, \underline{\mathbf{G}}_k^{(X)}$  for  $k = 1, \dots, K$ ;  $n = 2, \dots, N$ 
1: Initialisation of  $\underline{\mathbf{E}}_1 = \underline{\mathbf{X}}$  and  $\mathbf{f}_1 = \mathbf{y}$ 
2: for  $k = 1$  to  $K$  do
3:   if  $\|\underline{\mathbf{E}}_k\| > \epsilon$  and  $\|\mathbf{f}_k\| > \epsilon$  then
4:      $\underline{\mathbf{C}}_k = \langle \underline{\mathbf{E}}_k, \mathbf{f}_k \rangle_{\{1,1\}}$ 
5:      $\underline{\mathbf{G}}_k^{(X)}, \mathbf{t}_k, \mathbf{P}_k^{(2)}, \dots, \mathbf{P}_k^{(N)} = ACE(\underline{\mathbf{E}}_k, \mathbf{f}_k)$  or  $ACCoS(\underline{\mathbf{E}}_k, \mathbf{f}_k)$ 
6:      $\mathbf{b}_k = \mathbf{t}_k \mathbf{f}_k$ 
7:      $\underline{\mathbf{E}}_{k+1} = \underline{\mathbf{E}}_k - [\underline{\mathbf{G}}_k^{(X)}; \mathbf{t}_k, \mathbf{P}_k^{(2)}, \dots, \mathbf{P}_k^{(N)}]$ 
8:      $\mathbf{f}_{k+1} = \mathbf{f}_k - \mathbf{t}_k \mathbf{b}_k$ 
9:   else
10:    break
11:   end if
12: end for

```

Figure 4.6: BTTR

Given a set of training data $\underline{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and vectoral response $\mathbf{y}_{\text{train}} \in \mathbb{R}^{I_1}$, BTTR training consists of automatically identifying K blocks s.t.

$$\underline{\mathbf{X}}_{\text{train}} = \sum_{k=1}^K \underline{\mathbf{G}}_k \times_1 \mathbf{t}_k \times_2 \mathbf{P}_k^{(2)} \times_3 \dots \times_N \mathbf{P}_k^{(N)} + \underline{\mathbf{E}}_k$$

$$\mathbf{y}_{\text{train}} = \sum_{k=1}^K \mathbf{u}_k + \mathbf{f}_k \text{ with } \mathbf{u}_k = \mathbf{t}_k b_k$$

with $\underline{\mathbf{G}}_k \in \mathbb{R}^{1 \times R_2^k \times \dots \times R_N^k}$ the core tensor for the k th-block, $\mathbf{P}_k^{(n)}$ the k th loading matrix for the n -mode, \mathbf{u}_k and \mathbf{t}_k the score vectors, b_k the regression coefficient, and $\underline{\mathbf{E}}_k$ and \mathbf{f}_k the residuals. Once the model is trained – and, hence, $\underline{\mathbf{G}}_k$, $\mathbf{P}_k^{(n)}$ and b_k are computed – the final prediction is obtained as: $\mathbf{y}_{\text{test}} = \mathbf{T}\mathbf{b} = \underline{\mathbf{X}}_{\text{test}(1)}\mathbf{W}\mathbf{b}$ where each column $\mathbf{w}_k = (\mathbf{P}_k^{(n)} \otimes \dots \otimes \mathbf{P}_k^{(2)})\text{vec}(\underline{\mathbf{G}}_k)$.

1) Generalized Block-Term Tensor Regression (gBTTR)

One possible limitation of BTTR is the lack of generalization as it aims to maximize the correlation between $\underline{\mathbf{X}}_{\text{train}}$ and $\mathbf{y}_{\text{train}}$ only, and this can lead to overfitting. Starting from the above prediction equation, we can re-write the above as the well-known linear regression equation $\mathbf{y}_{\text{test}} = \underline{\mathbf{X}}_{\text{test}(1)}\mathbf{b}_{\text{BTTR}}$ with $\mathbf{b}_{\text{BTTR}} = \mathbf{W}\mathbf{b}$.

In the context of linear regression, it has been shown that by randomly adding Gaussian noise $\epsilon \sim (0, \sigma)$ to the inputs increases the robustness of the model against overfitting while improving its generalization [48]. More specifically, the process consists of fitting the linear regression to the manipulated noisy data. Clearly, different regression lines are obtained each time random noise is added to the data. Therefore, to obtain a stable result, a process called marginalization is introduced to integrate out such randomness. In practice, marginalization consists of averaging the estimated regression lines. Such averaged regression line with noise σ is equal to the ridge regression line with penalty parameter $\lambda = N\sigma^2$ [48] with N the number of observations in the data set. We adopt the aforementioned idea as follows: At each iteration k , \mathbf{f}_k is manipulated M times by adding random Gaussian noise; for each \mathbf{f}_k^M the corresponding $\mathbf{w}_k^M \mathbf{b}_k^M$ solution is determined; then, we compute the averaged solution (marginalization), $\hat{\mathbf{w}}_k \hat{\mathbf{b}}_k$, followed by deflation before the next iteration. The final prediction is $\mathbf{y}_{\text{test}} = \mathbf{X}_{\text{test}(1)} \hat{\mathbf{b}}_{\text{BTTR}}$ where the k th element in $\hat{\mathbf{b}}_{\text{BTTR}}$ is computed via marginalization in the k th iteration, i.e. $\hat{\mathbf{w}}_k \hat{\mathbf{b}}_k$. We will refer to this ridge-like approach as generalized BTTR (gBTTR).

4.4.3 Time Complexity Analysis

Given training data $\underline{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and vectoral response $\mathbf{y}_{\text{train}} \in \mathbb{R}^{I_1}$ with $I_1 = I_2 = \dots = I_N$ and $R_1 = R_2 = \dots = R_N$, gBTTR has approximately time complexity $O(I^{(N)} R + R^{3(N-1)} + R^{2(N-1)} I)$. For comparison, HOPLS, under the same circumstances, has time complexity $O(I^{(N+M)} R^2 + R^{4(N+M-2)} + R^{3(N+M-2)} I)$ [49].

4.4.4 Non-multilinear approaches

As multilinear approaches are less common in BCI, we will consider the BCI competition IV dataset (see further) for comparing the performance of BTTR and HOPLS with the winner of this competition a linear regression model based on amplitude modulation (AM) [22], as well as with more recent attempts based on Random Forests (RF), Convolutional Neural Networks (CNN), and Long Short-Term Memory Network (LSTM) (proposed and compared in [23]).

Since in both [22] and [23] only 1 test set was used to assess performance, which factually impedes statistically testing the significance of any observed performance difference between the compared algorithms, we have re-implemented AM, RF, LARS, CNN and LSTM. We proceeded as follows.

For AM, we replicated the procedure described in [22]. First, we filtered the ECoG signals in the sub-gamma (1-60Hz), gamma (60-100 Hz) and high-gamma bands (100-200 Hz). For each of these bands, we determined the amplitude

modulation and used it to estimate their band-specific AM features. For each finger and subject, we used forward feature selection using a wrapper approach to find the relevant AM features. These features are then used in a linear regression model. Finally, we verified whether the obtained performance compared with the one reported in [22].

For LARS, we used the *LassoLars* function from the most recent version of the *scikit-learn* package of Python (version 0.24.2 released in April 2021; we assume that Xie et al. used version 0.19.1, but this was not reported). Hence, small differences in performance could be occur. LARS transforms the original signal using ICA, decomposes it into different bands, calculates band powers and fits a LassoLars model. LassoLars has one main parameter, α , the multiplier for the penalty term whereby $\alpha = 0$ corresponds to ordinary least square linear regression. A line search was used to optimize α .

For RF, a similar processing pipeline was used. The *RandomForestRegressor* function from the *scikit-learn* package was used (same version as above). RF transforms the original signal using ICA, decomposes it into different bands, calculates band powers and fits the *RandomForestRegressor* model.

The CNN and LSTM were built according to the specifications and architecture given in [23]. CNN refers to a linear regression model applied to features prior extracted using a CNN. The CNN and LSTM were build using PyTorch in Python (version 1.8.0 released in March 2021; note that Xie et al. did not specify which version they used).

Temporal SVM-rbf and Temporal lightGBM, introduced in [27] use Riemannian-space features and temporal dynamics of the ECoG signal combined with modern machine learning.

4.5 Data Sets Used

We first use synthetic data to compare state-of-the-art methods of automatic MTR determination versus the proposed ACE and ACCoS. Then, we adopt a real-world scenario for BTTR, HOPLS and several linear regression- and deep learning methods: decoding finger movement trajectories from ECoG recordings in humans, a challenging goal in the BCI community due to the presence of irrelevant and nonstationary brain activity and noise.

4.5.1 Synthetic Data For Component Analysis

The synthetic data is created for the predictor N th-order tensor variable $\underline{\mathbf{X}} = \underline{\mathbf{X}}_c + (1 - \alpha)\underline{\mathbf{X}}_u \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and the response vector variable $\mathbf{y} \in \mathbb{R}^{I_1 \times 1}$. The response variable \mathbf{y} is randomly generated. The $\underline{\mathbf{X}}_u \sim \mathcal{N}(0, 1)$ is an uncorrelated

tensor with randomly generated entries while $\underline{\mathbf{X}}_c$ is the true correlated tensor, defined according to the Tucker model as $\underline{\mathbf{X}}_c = [\underline{\mathbf{G}}, \mathbf{t}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}]$ with \mathbf{t} , the score vector and the factor matrices, and $\underline{\mathbf{G}} \sim \mathcal{N}(0, 1) \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$ the randomly generated core tensor. Note that $R_1 = 1$. The entries (r_n, i_n) of the loading factor $\mathbf{P}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ are defined as

$$\mathbf{P}_r^{(n)} \begin{cases} \sin(2\pi r_2 \frac{i_2}{I_2}) & \text{for } n = 2 \\ \cos(2\pi r_3 \frac{i_3}{I_3}) & \text{for } n = 3 \\ \operatorname{sgn}(\sin(2\pi r_4 \frac{i_4}{I_4})) & \text{for } n = 4 \end{cases}$$

with $r_n = 1, \dots, R_n$ and $i_n = 1, \dots, I_n$, whereas the score factor $t \in \mathbb{R}^{I_1 \times 1}$ is set equal to \mathbf{y} . The parameter α defines the weight of $\underline{\mathbf{X}}_u$ on the final tensor $\underline{\mathbf{X}}$ and, therefore, regularizes the correlation level between $\underline{\mathbf{X}}$ and \mathbf{y} with $\alpha = 1$ and $\alpha = 0$ yielding, respectively, maximal and minimal correlation.

We performed a series of tests to evaluate the robustness of the proposed ACE model as well as mPSTD and ACCoS under different conditions and compared their performances with the state-of-the-art models in automatic rank determination (see Previous work on automatic multilinear tensor rank selection). Several tests are performed by varying data properties aimed at showing how they affect the recovery of the true correlated (hidden) components. We chose $N=4$ and varied one property at a time while keeping the others constant. The data properties we explored are listed below:

- Correlation level: parameter α is chosen as 0.1, 0.5 or 1 to regularize the interference of uncorrelated sources.
- Number of components (i.e. MTR): parameter set R_2, \dots, R_N defines the size of the (correlated) factors $\mathbf{P}^{(n)} \in \mathbb{R}^{I_n \times R_n}_{n=2}^N$, i.e., the components of interest, and are heuristically chosen as [1, 1, 1], [2, 2, 1], [2, 1, 2], [1, 2, 2], or [2, 2, 2].
- Tensor dimensionality: parameter set $I_2, \dots, I_N \in [5, 5, 5], [10, 10, 10]$, or [20, 20, 20] defines the dimensionality of the Nth-order independent variable $\underline{\mathbf{X}}$.
- Sample size: parameter $I_1 \in 100, 1000$, or 10000 defines the number of observations.

We used two values to assess performance: how accurately the expected true components are retrieved (C1) and how much of the relevant information is contained in the extra components (C2). Note that we report NA for C2 when no extra components are available. We repeated each test 10 times and for each run the data was randomly generated as described above. Finally, we reported C1 and C2 in terms of mean and standard deviation as well as number of extra components averaged across modes.

4.5.2 Regression Analysis Of Real-Word Data: Decoding Finger Movement Trajectories From ECoG Signals Recorded in Humans

We also compared the proposed BTTR versions with HOPLS as well as several conventional linear regression approaches and deep learning networks in a real-word application. We adopted the dataset used in BCI competition IV where the task was to predict continuous finger flexions from ECoG signals recorded from the motor cortex, sampled at 1000 Hz. Three subjects were cued to move a particular finger at a particular moment (gauged with a data glove). In total, 150 trials were executed (30 trials per finger) in a single session lasting 600 seconds. For each trial, subjects typically flexed the cued finger 3–5 times for 2 seconds followed by a rest period of 2 seconds. The first 400 and the last 200 seconds of recording were used as training and testing set respectively. The number of ECoG channels varied across subjects. More details about the data can be found in [36].

1) Data preparation for multiway analysis

First, the ECoG and data glove signals were preprocessed. The data glove data was, independently for each finger, normalized (z-scores) yielding a vector $y \in \mathbb{R}^{\text{Samples}}$ for each finger. The ECoG signal was first inspected to identify and further exclude bad channels: we found that channels 55 in subject 1, 21 and 38 in subject 2, and 50 in subject 3 were affected by strong artifacts and therefore removed. The remaining channels were re-referenced using the common average reference (CAR) technique [50]. Then, using bidirectional fourth-order Butterworth band-pass filters, the ECoG signals were subjected to 8 band-pass filters, as commonly done in ECoG based BCI [37]: δ (1.5 -5 Hz), θ (5 -8 Hz), α (8 -12 Hz), β_1 (12 -24 Hz), β_2 (24 -34 Hz), γ_1 (34 -60 Hz), γ_2 (60 - 100 Hz), γ_3 (100 - 130 Hz). For each cued finger flexion trial, the glove data and the bandpass filtered ECoG signals were extracted starting 1 sec prior to trial onset ("epoch"). ECoG epochs were downsampled to 10 Hz to further reduce data size. All epochs were then concatenated into a unique fourth-order tensor $\underline{X} \in \mathbb{R}^{\text{Samples} \times \text{Channels} \times 8(\text{Frequencies}) \times 10(\text{Time})}$. There were 62, 48, and 64 ECoG channels for subjects 1, 2, and 3, respectively. Finally, \underline{X} is normalized (z-scores) to reduce the difference in magnitude between frequency bands. A graphical representation of the data preparation is shown in Figure 4.7.

2) Calibration, validation, performance assessment

A 5-fold cross-validation approach was used to optimize the model parameters on the training data, i.e., K, R_2, \dots, R_N for HOPLS and K for BTTR. In addition, in support of the statistical analysis, the test data was split into 5 non-overlapping

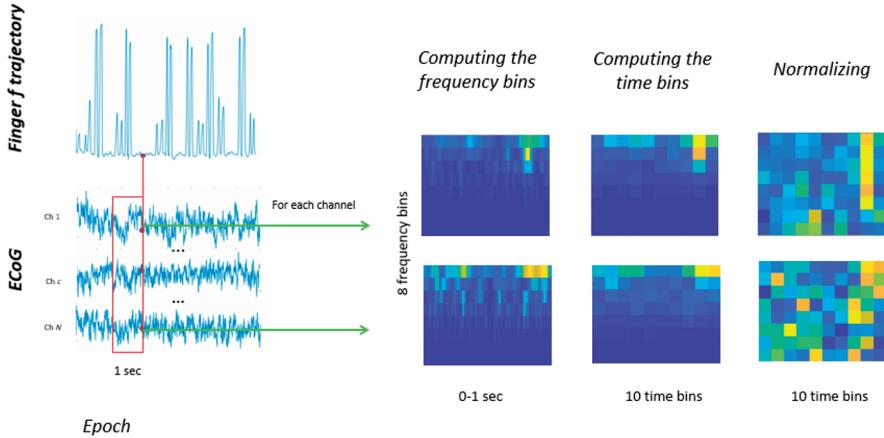


Figure 4.7: Diagram with the steps to create an ECoG tensor.

blocks. We also proceed in this way for the winning algorithm of the BCI competition IV, as well as its more recent attempts.

The decoding performance for each block was measured using Pearson's correlation coefficient between predicted and actual movement trajectory, which are then averaged. We used a two-tailed Wilcoxon signed-rank test [51] to compare the models for each subject. Two results are considered significantly different if the p-value is < 0.05 .

4.6 Results

In this section, we compare the results obtained for the state-of-the-art methods of automatic MTR selection and the proposed ACE and ACCoS for the case of synthetic data, under different conditions. Then we compare the predictive performance of the proposed multiway regression, BTTR, with and without automatic component selection, for a real-world scenario.

4.6.1 ACE And ACCoS On Synthetic Data For Component Extraction

For the synthetic data case, we performed various tests to verify model robustness under different conditions such as 1) correlation level between \underline{X} and y , 2) rank size, 3) tensor dimensionality, and 4) sample size. We first show the efficiency of the proposed ACE against manual tuning of model parameters required for

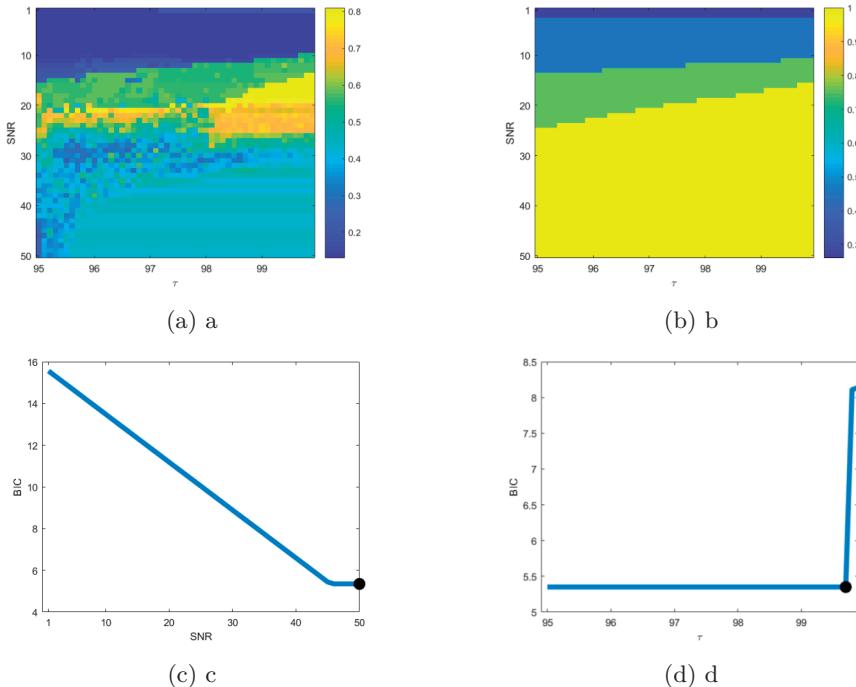


Figure 4.8: Top left panel (Standard PSTD): C1 accuracy (color coded, see scale on the right) achieved by standard PSTD when (manually) varying model parameters SNR and τ . Top right panel (mPSTD): idem but for the proposed mPSTD. Bottom left (BIC-SNR) and right (BIC-Ratio for best SNR): filled circles indicate the selected SNR (50) and τ (99.7) values.

the proposed mPSTD and standard PSTD. Then we compare ACE and ACCoS with state-of-the-art methods under the same conditions.

In Figure 4.8 we show an example of mPSTD and standard PSTD (top panels) as well the outcome of the BIC analysis included in ACE for the automatic selection of model parameters SNR and τ (bottom panels) when $\alpha = 0.5$, $R_2, R_3, R_4 = [2, 2, 2]$, $I_1 = 10000$, and $I_2, I_3, I_4 = [20, 20, 20]$. Initialization of the factor matrices seems to be a key element in extracting the correct components and in overall convergence. This is not noticed when comparing mPSTD and standard PSTD (Figure 4.8, top panel). The proposed mPSTD tends to be less sensitive to the choice of model parameters as for standard PSTD, which relies on random initialization, only few SNR- τ combinations lead to a correct solution. Interestingly, mPSTD generally yields a more accurate

solution ($\max C_1 = 1$) than standard PSTD ($\max C_1 = 0.81$). In addition, the outcome of this study shows that ACE, as it relies on the BIC approach, is able to determine the optimal model parameters (leading to $C_1 = 1$) under any condition in a fully automatic way. An obvious advantage of the proposed ACE is the automatic selection of the parameters –generally obtained via cross-validation– and the absence of any prior assumptions.

Additional tests showed that both ACCoS and ACE perform at least as good as the other state-of-the-art methods TREL1 and SCORE as they are able to retrieve, in a fully automatic way, the true correlated components regardless of the properties of the data (cf., C1 result). However, ACCoS differs from the others as it can successfully identify and reject irrelevant components (cf., C2 result). An example is reported in Figure 4.9 where the correlation level between the two variables changes while fixing $R_2, R_3, R_4 = [2, 2, 2]$, $I_1 = 10000$, and $I_2, I_3, I_4 = [20, 20, 20]$.

	ACE		ACCoS		TREL1		Tucker ARD		SCORE	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
.1	1	0.2 (1)	1	NA	1	0.2 (8)	0.2	0.2 (8.7)	1	0.2 (1)
.5	1	0.2 (1)	1	NA	1	0.2 (8)	0.7	0.2 (8)	1	0.2 (1)
1	1	NA	1	NA	1	NA	0.8	0.3 (8)	0.5	0.4 (8.6)

Figure 4.9: Results of correlation analysis when varying the correlation parameter α while keeping $R_2, R_3, R_4 = [2, 2, 2]$, $I_1 = 10000$, and $I_2, I_3, I_4 = [20, 20, 20]$. The reported values represent the correctness of the expected components (C1) and the extra components (C2 with average of number of extra components in brackets).

4.6.2 BTTR on Real-World Data: Decoding Finger Movement Trajectories from Human ECoG Recordings

For clarity’s sake, we first compared the proposed versions of BTTR against HOPLS to select the best BTTR version on a real-world case, then compared this version against the BCI competition IV winner and the more recent attempts. We analyzed for each subject one finger at a time. Figure 4.10 shows an example of predicted finger movement for gBTTR and HOPLS together with the data glove signal.

1) Comparison of multiway regression approaches: BTTR versions against HOPLS

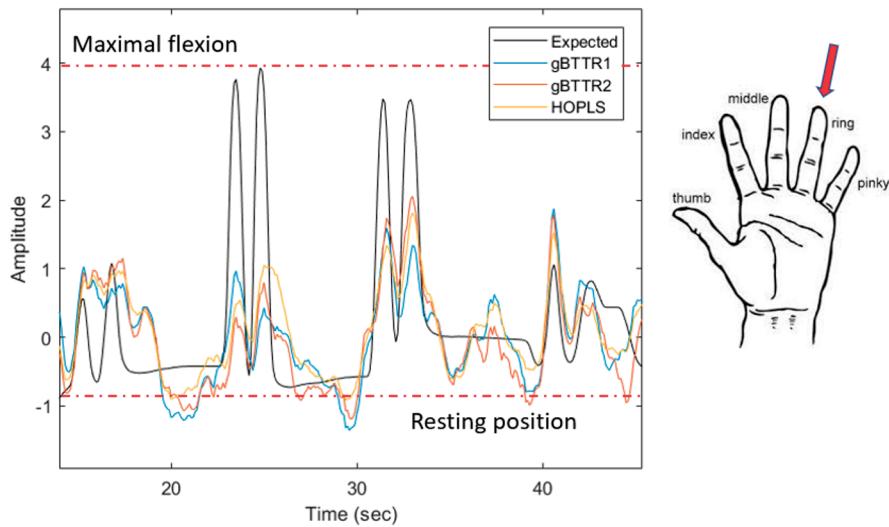


Figure 4.10: Predicted ring finger flexions of Subject 1. Actual (data glove) and predicted (regression models) finger flexion amplitudes (z-scores) are plotted as function of time (in ms). Curve colors are explained in the inset. For the sake of exposition, only results for the generalized version gBTTR are shown.

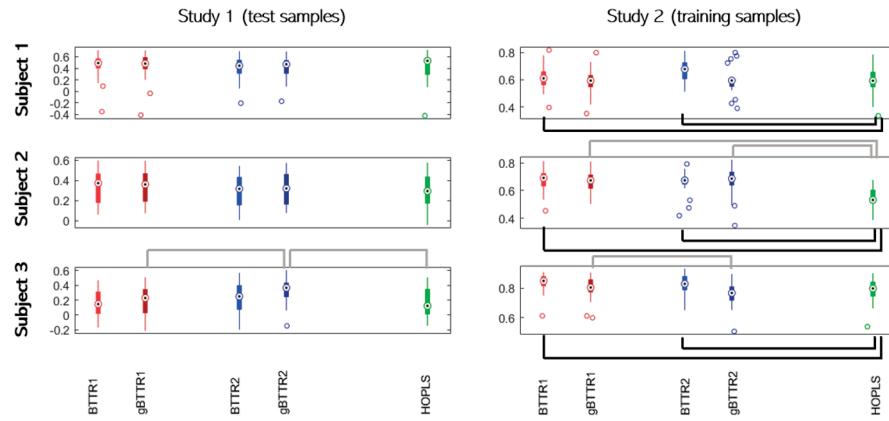


Figure 4.11: Performances of the proposed models and HOPLS when predicting training and test data.

In Figure 4.11, we show for each subject (panels shown row-wise) the averaged (across fingers) correlation coefficients obtained when predicting test samples (Study 1, panels in left column) and when predicting samples used for model training (Study 2, panels in right column).

When analyzing the results of Study 1, statistical analysis (see Calibration, validation, performance assessment) revealed no significant difference between the performance of BTTR and that of its generalized version gBTTR. However, ACCoS-gBTTR performs better compared to ACE-gBTTR and HOPLS for subject 3 (gray brackets).

In Study 2, ACCoS-BTTR performs equally well as ACE-BTTR and both models provide better accuracies than HOPLS (black brackets, all subjects). When considering the gBTTR version, performances are lower than with BTTR. Indeed, except for subject 2 (gray brackets), the difference with HOPLS tends to disappear when using gBTTR.

To summarize, for ACCoS-gBTTR, prediction accuracy of the test data improved while that of its training data decreased: the predictions obtained with ACCoS-BTTR, ACE-BTTR and ACE-gBTTR are comparable to the ones obtained with HOPLS while for ACCoS-gBTTR they are at least as good. Hence, seemingly, the generalized version of BTTR is able to limit the effect of overfitting mainly for ACCoS-BTTR; indeed, ACE-BTTR extracts also components that are not of interest and that -as in the HOPLS case—demote prediction accuracy.

We further investigated the runtime required by these models to identify their parameters. Overall, ACE-BTTR and ACCoS-BTTR are trained faster than HOPLS (e.g. 3 minutes and 8 minutes respectively against 14 hours for HOPLS). This is in line with the time complexity found for the algorithms. Note that HOPLS requires computationally expensive techniques such as cross-validation to identify the optimal set of model parameters (i.e., the number of scores and loadings). In contrast, BTTR and gBTTR automatically determine the model parameters (i.e., the number of loadings) leading to a model that combines high flexibility with more natural representation of complex multiway data, although it still requires (a simple) cross-validation to determine the number of scores. The next comparison continues with ACE-BTTR as it is the most well-rounded of the various versions in terms of speed and performance.

2) Comparison with non-multiway approaches

We also compared the ACE-BTTR model and HOPLS with the winner of the BCI competition IV, linear regression of amplitude modulation (AM) [22], as well as 4 more recent attempts as the competition concluded some time ago: Random Forest (RF), Least Angle Regression (LARS), Convolutional

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
ACE-BTTR	0.72 ± .06	0.77 ± .09	0.38 ± .02	0.68 ± .04	0.67 ± .02	0.64 ± .05
HOPLS	0.70 ± .05	0.79 ± .08	0.36 ± .03	0.70 ± .06	0.65 ± .02	0.63 ± .04
AM	0.57 ± .03	0.69 ± .06	0.14 ± .02	0.52 ± .04	0.28 ± .01	0.42 ± .03
RF	0.58 ± .09	0.54 ± .05	0.07 ± .03	0.31 ± .05	0.33 ± .02	0.38 ± .05
LARS	0.11 ± .05	0.08 ± .03	0.10 ± .02	0.60 ± .05	0.39 ± .02	0.17 ± .03
CNN	0.67 ± .04	0.78 ± .04	0.11 ± .02	0.54 ± .03	0.45 ± .04	0.50 ± .04
LSTM	0.73 ± .03	0.79 ± .08	0.18 ± .02	0.61 ± .04	0.45 ± .04	0.54 ± .04
SVM-rbf	0.594	0.734	0.364	0.395	0.480	0.54
LightGBM	0.543	0.760	0.401	0.383	0.531	0.558

Table 4.2: Finger trajectory decoding performance (Pearson correlation) of BTTR and other models for Subject 1.

Neural Network (CNN), and Long Short-Term Memory Network (LSTM). The correlation coefficients are given for each finger individually and averaged across all fingers, except for finger 4 (ring) as flexing the latter is difficult to suppress when the 3rd or 5th finger is flexing. The average results and their standard deviations for the 5 blocks of test data (see above) are listed in Table 4.2, Table 4.3 and Table 4.4 (listed under Avg.) for subjects 1, 2 and 3, respectively.

For all 3 subjects, a statistically significant difference (see Calibration, validation, performance assessment) in average results was found between ACE-BTTR and LSTM and between AM/RF/LARS and ACE-BTTR. It is interesting to note that on a per-finger basis, there was not always a statistically significant difference between individual algorithms. For instance, for all 3 subjects, there is no statistical difference for the thumb and index finger between ACE-BTTR and LSTM. For subjects 1 and 3, there is a significant difference for the middle finger between ACE-BTTR and LSTM. For subjects 1 and 2, there is a significant difference for the pinky between ACE-BTTR and LSTM.

SVM-rbf and LightGBM [27] are the main state-of-the-art methodologies. ACE-BTTR performs better for subject 1 compared to these methodologies. LightGBM performs better compared to ACE-BTTR for subject 2. Looking at a per-finger case, the main difference is the middle finger which performs far better in LightGBM. For subject 3, ACE-BTTR performs equally to LightGBM with better decoding for middle, ring and thumb, but worse performance for the thumb and index finger.

4.7 Conclusion

Tensor techniques can be an important asset to BCI as they can outperform more conventional decoders in accuracy and reliability but their proliferation

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
ACE-BTTR	0.64 ± .05	0.46 ± .08	0.27 ± .04	0.50 ± .03	0.48 ± .01	0.46 ± .05
HOPLS	0.63 ± .04	0.47 ± .06	0.26 ± .05	0.51 ± .02	0.48 ± .01	0.44 ± .04
AM	0.52 ± .03	0.36 ± .06	0.23 ± .02	0.48 ± .04	0.33 ± .01	0.36 ± .03
RF	0.52 ± .05	0.36 ± .04	0.22 ± .03	0.39 ± .04	0.25 ± .02	0.34 ± .04
LARS	0.54 ± .05	0.41 ± .04	0.18 ± .02	0.44 ± .04	0.25 ± .02	0.35 ± .03
CNN	0.60 ± .04	0.40 ± .04	0.24 ± .02	0.44 ± .03	0.28 ± .04	0.38 ± .04
LSTM	0.62 ± .03	0.38 ± .08	0.27 ± .02	0.47 ± .04	0.30 ± .04	0.39 ± .04
SVM-rbf	0.591	0.481	0.344	0.421	0.437	0.463
LightGBM	0.658	0.522	0.387	0.392	0.407	0.493

Table 4.3: Idem to Table 4.2 but for Subject 2.

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
ACE-BTTR	0.73 ± .05	0.59 ± .08	0.64 ± .04	0.63 ± .02	0.72 ± .01	0.67 ± .05
HOPLS	0.74 ± .06	0.57 ± .09	0.65 ± .02	0.61 ± .04	0.68 ± .02	0.64 ± .04
AM	0.59 ± .03	0.51 ± .06	0.32 ± .02	0.53 ± .04	0.42 ± .01	0.46 ± .03
RF	0.67 ± .05	0.27 ± .04	0.16 ± .03	0.14 ± .04	0.36 ± .02	0.37 ± .04
LARS	0.72 ± .05	0.43 ± .04	0.45 ± .02	0.51 ± .04	0.64 ± .02	0.56 ± .03
CNN	0.74 ± .03	0.53 ± .05	0.45 ± .04	0.49 ± .03	0.68 ± .06	0.60 ± .05
LSTM	0.74 ± .02	0.55 ± .06	0.46 ± .04	0.41 ± .02	0.75 ± .06	0.62 ± .05
SVM-rbf	0.767	0.654	0.513	0.347	0.638	0.643
LightGBM	0.82	0.648	0.578	0.358	0.664	0.6775

Table 4.4: Idem to Table 4.2 but for Subject 3.

is hindered by the computational intensive parameter estimation. Model parameters are often optimized via time consuming techniques such as cross-validation on a sufficient set of parameter combinations. To tackle this limitation, we proposed a new tensor decomposition approach for regression that we enhanced with automatic rank determination and showed that it can challenge state-of-the-art multiway regression techniques while outperforming more conventional ones. The proposed solution is characterized by flexible modeling, supporting the representation of complex data, and by fast model training. This can open new perspectives for multiway data modeling in BCI applications.

Acknowledgment

AF was supported by the National Fund for Scientific Research – Flanders (FWO 1157019N). FC was supported by the Hermes Fund of the National Fund for Scientific Research – Flanders (IWT.141112). MMVH is supported by research grants received from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 857375, the special research fund of the KU Leuven (C24/18/098), the Belgian Fund for Scientific Research

- Flanders (G0A0914N, G0A4118N, G0A4321N), the Interuniversity Attraction Poles Programme — Belgian Science Policy (IUAP P7/11), and the Hercules Foundation (AKUL 043).

Bibliography

- [1] A. Vallabhaneni, T. Wang, and B. He, “Brain—computer interface”, in *Neural engineering*, Springer, 2005, pp. 85–121.
- [2] A. B. Ajiboye, F. R. Willett, D. R. Young, W. D. Memberg, B. A. Murphy, J. P. Miller, B. L. Walter, J. A. Sweet, H. A. Hoyen, M. W. Keith, *et al.*, “Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration”, *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, 2017.
- [3] C. E. Bouton, A. Shaikhouni, N. V. Annetta, M. A. Bockbrader, D. A. Friedenberg, D. M. Nielson, G. Sharma, P. B. Sederberg, B. C. Glenn, W. J. Mysiw, *et al.*, “Restoring cortical control of functional movement in a human with quadriplegia”, *Nature*, vol. 533, no. 7602, pp. 247–250, 2016.
- [4] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia”, *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.
- [5] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, “High-performance neuroprosthetic control by an individual with tetraplegia”, *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.
- [6] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simner, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt, *et al.*, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm”, *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.
- [7] B. Wodlinger, J. Downey, E. Tyler-Kabara, A. Schwartz, M. Boninger, and J. Collinger, “Ten-dimensional anthropomorphic arm control in a human brain- machine interface: Difficulties, solutions, and limitations”, *Journal of neural engineering*, vol. 12, no. 1, p. 016011, 2014.

- [8] W. Wang, J. L. Collinger, A. D. Degenhart, E. C. Tyler-Kabara, A. B. Schwartz, D. W. Moran, D. J. Weber, B. Wodlinger, R. K. Vinjamuri, R. C. Ashmore, *et al.*, “An electrocorticographic brain interface in an individual with tetraplegia”, *PLoS one*, vol. 8, no. 2, e55344, 2013.
- [9] E. S. Nurse, S. E. John, D. R. Freestone, T. J. Oxley, H. Ung, S. F. Berkovic, T. J. O’Brien, M. J. Cook, and D. B. Grayden, “Consistency of long-term subdural electrocorticography in humans”, *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 2, pp. 344–352, 2017.
- [10] K. J. Miller, L. B. Sorensen, J. G. Ojemann, and M. Den Nijs, “Power-law scaling in the brain surface electric potential”, *PLoS Comput Biol*, vol. 5, no. 12, e1000609, 2009.
- [11] R. J. Staba, C. L. Wilson, A. Bragin, I. Fried, and J. Engel Jr, “Quantitative analysis of high-frequency oscillations (80–500 hz) recorded in human epileptic hippocampus and entorhinal cortex”, *Journal of neurophysiology*, vol. 88, no. 4, pp. 1743–1752, 2002.
- [12] Z. C. Chao, Y. Nagasaka, and N. Fujii, “Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey”, *Frontiers in neuroengineering*, vol. 3, p. 3, 2010.
- [13] T. Ball, M. Kern, I. Mutschler, A. Aertsen, and A. Schulze-Bonhage, “Signal quality of simultaneously recorded invasive and non-invasive eeg”, *Neuroimage*, vol. 46, no. 3, pp. 708–716, 2009.
- [14] J. C. Siero, D. Hermes, H. Hoogduin, P. R. Luijten, N. F. Ramsey, and N. Petridou, “Bold matches neuronal activity at the mm scale: A combined 7 t fmri and ecog study in human sensorimotor cortex”, *Neuroimage*, vol. 101, pp. 177–184, 2014.
- [15] G. Schalk, K. J. Miller, N. R. Anderson, J. A. Wilson, M. D. Smyth, J. G. Ojemann, D. W. Moran, J. R. Wolpaw, and E. C. Leuthardt, “Two-dimensional movement control using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 5, no. 1, p. 75, 2008.
- [16] T. Yanagisawa, M. Hirata, Y. Saitoh, H. Kishima, K. Matsushita, T. Goto, R. Fukuma, H. Yokoi, Y. Kamitani, and T. Yoshimine, “Electrocorticographic control of a prosthetic arm in paralyzed patients”, *Annals of neurology*, vol. 71, no. 3, pp. 353–361, 2012.
- [17] E. A. Felton, J. A. Wilson, J. C. Williams, and P. C. Garell, “Elecrocorticographically controlled brain-computer interfaces using motor and sensory imagery in patients with temporary subdural electrode implants: Report of four cases”, *Journal of neurosurgery*, vol. 106, no. 3, pp. 495–500, 2007.
- [18] A. L. Benabid, T. Costecalde, A. Eliseyev, G. Charvet, A. Verney, S. Karakas, M. Foerster, A. Lambert, B. Morinière, N. Abroug, *et al.*, “An exoskeleton controlled by an epidural wireless brain-machine interface

- in a tetraplegic patient: A proof-of-concept demonstration”, *The Lancet Neurology*, vol. 18, no. 12, pp. 1112–1122, 2019.
- [19] J. Kubanek, K. J. Miller, J. G. Ojemann, J. R. Wolpaw, and G. Schalk, “Decoding flexion of individual fingers using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 6, no. 6, p. 066 001, 2009.
 - [20] Z. Wang, Q. Ji, K. J. Miller, and G. Schalk, “Decoding finger flexion from electrocorticographic signals using a sparse gaussian process”, in *2010 20th International conference on pattern recognition*, IEEE, 2010, pp. 3756–3759.
 - [21] R. Flamary and A. Rakotomamonjy, “Decoding finger movements from ecog signals using switching linear models”, *Frontiers in neuroscience*, vol. 6, p. 29, 2012.
 - [22] N. Liang and L. Bougrain, “Decoding finger flexion from band-specific ecog signals in humans”, *Frontiers in neuroscience*, vol. 6, p. 91, 2012.
 - [23] Z. Xie, O. Schwartz, and A. Prasad, “Decoding of finger trajectory from ecog using deep learning”, *Journal of neural engineering*, vol. 15, no. 3, p. 036 009, 2018.
 - [24] G. Pfurtscheller and C. Neuper, “Motor imagery and direct brain-computer communication”, *Proceedings of the IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.
 - [25] E. Acar and B. Yener, “Unsupervised multiway data analysis: A literature survey”, *IEEE transactions on knowledge and data engineering*, vol. 21, no. 1, pp. 6–20, 2008.
 - [26] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons, 2005.
 - [27] L. Yao, B. Zhu, and M. Shoaran, “Fast and accurate decoding of finger movements from ecog through riemannian features and modern machine learning techniques”, *Journal of Neural Engineering*, vol. 19, no. 1, p. 016 037, 2022.
 - [28] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis”, *IEEE signal processing magazine*, vol. 32, no. 2, pp. 145–163, 2015.
 - [29] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications”, *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
 - [30] L. De Lathauwer, “Decompositions of a higher-order tensor in block terms—part i: Lemmas for partitioned matrices”, *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.
 - [31] ——, “Decompositions of a higher-order tensor in block terms—part ii: Definitions and uniqueness”, *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.

- [32] L. De Lathauwer and D. Nion, “Decompositions of a higher-order tensor in block terms—part iii: Alternating least squares algorithms”, *SIAM journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.
- [33] B. Hunyadi, D. Camps, L. Sorber, W. Van Paesschen, M. De Vos, S. Van Huffel, and L. De Lathauwer, “Block term decomposition for modelling epileptic seizures”, *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–19, 2014.
- [34] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, “Higher order partial least squares (hopls): A generalized multilinear regression method”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1660–1673, 2012.
- [35] A. Eliseyev and T. Aksanova, “Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording”, *PloS one*, vol. 11, no. 5, e0154878, 2016.
- [36] K. J. Miller and G. Schalk, “Prediction of finger flexion: 4th brain-computer interface data competition”, *BCI Competition IV*, vol. 1, pp. 1–2, 2008.
- [37] L. R. Tucker, “Some mathematical notes on three-mode factor analysis”, *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [38] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition”, *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [39] S. Zubair and W. Wang, “Tensor dictionary learning with sparse tucker decomposition”, in *2013 18th international conference on digital signal processing (DSP)*, IEEE, 2013, pp. 1–6.
- [40] A. H. Phan and A. Cichocki, “Fast and efficient algorithms for nonnegative tucker decomposition”, in *International Symposium on Neural Networks*, Springer, 2008, pp. 772–782.
- [41] L. De Lathauwer, B. De Moor, and J. Vandewalle, “On the best rank-1 and rank-(r₁, r₂, ..., r_n) approximation of higher-order tensors”, *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.
- [42] M. Mørup and L. K. Hansen, “Sparse coding and automatic relevance determination for multi-way models”, in *SPARS’09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.
- [43] T. Yokota and A. Cichocki, “Multilinear tensor rank estimation via sparse tucker decomposition”, in *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, IEEE, 2014, pp. 478–483.

- [44] T. Yokota, N. Lee, and A. Cichocki, “Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria”, *IEEE Transactions on Signal Processing*, vol. 65, no. 5, pp. 1196–1206, 2016.
- [45] Q. Shi, H. Lu, and Y.-m. Cheung, “Tensor rank estimation and completion via cp-based nuclear norm”, in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 949–958.
- [46] G. I. Allen and M. Maletić-Savatić, “Sparse non-negative generalized pca with applications to metabolomics”, *Bioinformatics*, vol. 27, no. 21, pp. 3029–3035, 2011.
- [47] G. Schwarz *et al.*, “Estimating the dimension of a model”, *Annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.
- [48] C. M. Bishop, “Training with noise is equivalent to tikhonov regularization”, *Neural computation*, vol. 7, no. 1, pp. 108–116, 1995.
- [49] F. Camarrone and M. M. Van Hulle, “Fast algorithm for multiway regression”, in *2017 22nd International Conference on Digital Signal Processing (DSP)*, 2017, pp. 1–5. DOI: [10.1109/ICDSP.2017.8096042](https://doi.org/10.1109/ICDSP.2017.8096042).
- [50] C. D. Binnie, *Clinical neurophysiology: EEG, paediatric neurophysiology, special techniques and applications*. Elsevier Health Sciences, 2003, vol. 2.
- [51] F. Wilcoxon, “Individual comparisons by ranking methods”, in *Breakthroughs in statistics*, Springer, 1992, pp. 196–202.

Chapter 5

Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression

AXEL FAES, AND MARC M. VAN HULLE

Journal of Neural Engineering 19.6 (2022): 066011.

5.1 Abstract

Objective We introduce extended Block-Term Tensor Regression (eBTTR), a novel regression method designed to account for the multilinear nature of human intracranial finger movement recordings.

Approach The proposed method relies on recursive Tucker decomposition combined with automatic component extraction.

Main results eBTTR outperforms state-of-the-art regression approaches, including multilinear and deep learning ones, in accurately predicting finger trajectories as well as unintentional finger coactivations.

Significance eBTTR rivals state-of-the-art approaches while being less computationally expensive which is an advantage when intracranial electrodes are implanted acutely, as part of the patient's presurgical workup, limiting time for decoder development and testing.

5.2 Introduction

Every year, up to half a million patients worldwide face paralysis due to spinal cord injury, brainstem stroke and amyotrophic lateral sclerosis (ALS) [1]. Brain Computer Interfaces (BCIs) are capable of bypassing disconnected neural pathways to replace the function of a lost or impaired body part, which led them to becoming promoted as a solution for these patients. Typically, BCI systems consist of several components: from the recorded brain activity, signal features are extracted and the result translated ("decoded") into commands controlling an external device such as a robotic arm or hand. Extraordinary results have been achieved with BCI controlling functional electrical stimulation (FES) of hand muscles [2], [3] and prosthetic hands and arms, exoskeletons or other effectors [4]–[7].

Motor BCI researchers are increasingly paying attention to electrocorticography (ECoG) where electrodes are placed directly above the cortex (i.e., sub- or epidurally) thereby avoiding tissue damage and histological processes wire microelectrode implants are prone to. ECoG signals yield significantly higher amplitudes compared to scalp EEG, are not contaminated by artifacts, enjoy a broader bandwidth and higher spatial resolution[8], [9], as well as long-term signal stability [10], [11].

Ramsey and co-workers at UMC Utrecht [12] reported success on the real-time, long-term use of an ECoG-based motor BCI in a patient with late-stage ALS. The patient imagined closing the right hand to select a cued target (i.e. a displayed character), while the decoder was detecting changes in the ECoG spectrum. Benabid et al. [13] reported a major breakthrough with a tetraplegic patient controlling a four-limb exoskeleton with up to 8 degrees of freedom (start/stop walking, arm trajectory, arm- and wrist rotations). Despite this success, Benabid et al. admitted that higher resolution ECoG electrode grids and further algorithmic developments are required to achieve better skilled movements of the joints and in particular of the hand (such as prehension and faster movements of the fingers). When looking beyond movement/no-movement classification, decoding finger movement trajectories

from ECoGs remains an unsolved problem, even though a few attempts have been made. For example, Kubanek et al. [14] were the first to continuously decode performed flexions/extensions of individual fingers by analyzing local motor potentials (LMP) and spectral amplitudes in 5 frequency bands. The authors used a conventional sparse linear regression model and obtained an average decoding performance of $r = 0.52$ (predicted vs. expected correlation). Since then, continuous finger movement decoding has been studied based on gaussian [15], linear regression [16], [17], Convolutional Neural Networks (CNN), Random Forests (RF) and deep learning networks (proposed in [18]). However, finger trajectory decoding could benefit from recent developments in multilinear algebra (multiway decoding) as ECoG is in essence structured in space, time, and frequency domains. This structure is largely ignored in traditional vector- or matrix-based regression models, as the data is concatenated into vectors and matrices (aka unfolding). Multiway models preserve the multilinear structure of the data and support the discovery of potentially hidden multilinear components [19]. The two most popular multiway frameworks are Tucker (TKD) and CANDECOMP/PARAFAC (or CPD) decompositions (see [19], [20] for review) and both aim to determine the low-dimensional space where important information is residing.

De Lathauwer introduced a tensor decomposition algorithm, called Block Term Decomposition (BTD) [21]–[23], which can be seen as a unified version of TKD and CPD. More specifically, in [23] two BTD models were proposed to approximate N^{th} -order tensor data as a sum of K-blocks called rank- (L_1^k, \dots, L_N^k) BTD and rank- (L_1, \dots, L_N) BTD with $k=1, \dots, K$. In the former model, blocks can have different multilinear rank (MTR), where in the latter a unique MTR is chosen for all blocks. If there is only one block, then BTD reduces to TKD. In contrast, if all blocks have multilinear rank MTR = $(1, 1, \dots, 1)$, then it reduces to CPD. This approach provides great flexibility and opens new possibilities for multiway data analysis, above all when rank- (L_1^k, \dots, L_N^k) BTD is adopted. For instance, an EEG study [24] showed that, while CPD failed to model epileptic seizures, rank- (L_1, \dots, L_N) BTD correctly extracted the ictal sources that matched clinical assessment.

The abovementioned multiway approaches have been adapted for regression analysis to model the relationship between arm trajectory and ECoG signals recorded from monkeys [25], [26] and recently between exoskeleton-based arm trajectory, arm- and wrist rotations and ECoG signals recorded from the tetraplegic patient mentioned above [13]. The decoder in the latter case relies on a variant of multiway partial least squares regression (NPLS), but is not capable of achieving the kind of accuracy needed to decode fine limb movements. Possible reasons for this underperformance are the limited fitness ability, the high computational complexity and the slow convergence of NPLS when handling

higher-order data [25]. On the other hand, Zhao et al. [25] developed a powerful generalized framework, called Higher-Order Partial Least Squares (HOPLS), based on $(1, L_2, \dots, L_N)$ -rank BTD (i.e. all blocks have the same MTR), that provides enhanced predictability with optimal balance between fitness and model complexity. As a result, HOPLS was able to outperform conventional PLS.

Camarrone and co-workers recently introduced Block-Term Tensor Regression (BTTR) [27], [28] and showed it performed with similar accuracy as HOPLS while being much faster to train. However, unlike BTTR, HOPLS generalises to predict a tensor (multiway array) \mathbf{Y} from a tensor \mathbf{X} by projecting the data onto latent space and performing regression on the corresponding latent variables. We introduce a similar generalization but for Block-Term Tensor Regression, further called eBTTR, while retaining its computational advantage over HOPLS and apply it to accurately decode finger movements, even unintended ones due to limited finger independence, further referred to as finger coactivation.

5.3 Materials and Methods

One of the main limitations of BTTR is that it can predict a scalar variable only. This becomes an issue when modeling coordinated finger movements as it in the case of grasping a cup. One evident solution is to have a BTTR model per finger. However, this limits exploiting information shared between fingers. More fundamentally, unlike individual fingers being represented at separate anatomical locations [29], when flexing multiple fingers simultaneously, the same locations now exhibit spatially sparse and even mixed signals [30]. This implies that decoders trained on individual finger movements could become inadequate when representing coordinated finger movements.

To tackle this, we extend BTTR into eBTTR to model multiple variables simultaneously. As an example case we consider intended single finger movement in the presence of unintended coactivation of the other fingers. An overview of the mathematical notation used can be found in Table 5.1.

Conceptually, the (e)BTTR algorithm consists of the following operations. First, the data is cast into tensor format, \mathbf{X} , with the corresponding “labels” into matrix format, \mathbf{Y} . (e)BTTR proceeds iteratively, yielding a series of blocks with declining contribution to the regression performance: at each iteration, a mode-1 cross-covariance tensor \mathbf{C} is constructed by combining information from both \mathbf{X} and \mathbf{Y} . This cross-covariance tensor \mathbf{C} is then decomposed and ACE (Automatic Parameter Extraction) used for automatic parameter estimation of the blocks. The resulting decomposition can be used to reconstruct \mathbf{X} and \mathbf{Y} and thus comprise the learned regression model (from this iteration). \mathbf{X} and \mathbf{Y}

Table 5.1: Mathematical notation

Notation	Description
$\underline{\mathbf{T}}, \mathbf{M}, \mathbf{v}, S$	tensor, matrix, vector, scalar (respectively)
\mathbf{M}^T	transpose of matrix
\times_n	mode-n product between tensor and matrix
\otimes	Kronecker product
\circ	outer product
$\ \cdot\ _F$	Frobenius norm
$\mathbf{T}_{(n)}$	mode-n unfolding of tensor $\underline{\mathbf{T}}$
$\underline{\mathbf{C}}^{(T)}$	core tensor associated to tensor $\underline{\mathbf{T}}$
$\mathbf{M}^{(n)}$	mode-n factor matrix
\mathbf{M}_{ind}	(sub-)matrix including the column(s) indicated in <i>ind</i>
$\mathbf{M}_{\backslash ind}$	(sub-)matrix excluding the column(s) indicated in <i>ind</i>
$[\![\underline{\mathbf{C}}; \mathbf{M}^{(1)}, \dots, \mathbf{M}^{(N)}]\!]$	full multilinear product $\underline{\mathbf{C}} \times_1 \mathbf{M}^{(1)} \times_2 \dots \times_N \mathbf{M}^{(N)}$
$\langle \underline{\mathbf{T}}, \underline{\mathbf{E}} \rangle_{\{n,n\}}$	mode-n cross-covariance tensor

get deflated before the next iteration starts. After training, the learned model can be used to predict \mathbf{Y} using a $\underline{\mathbf{X}}_{test}$.

Similar to HOPLS, BTTR and its multivariate extension eBTTR rely on a partial least squares approach (PLS). Due to this, their performance is similar to that of HOPLS. Conceptually, the main differences with HOPLS are: the decomposition of the mode-1 cross-product between predictor- and response variables, the blocks that can have different multilinear ranks (MTRs), and the use of the Automatic Component Extraction (ACE) for parameter estimation. These differences cause (e)BTTR to be faster to train than HOPLS, with comparable accuracies.

5.3.1 Extended Block-Term Tensor Regression (eBTTR)

The proposed extended Block-Term Regression (eBTTR) model is based

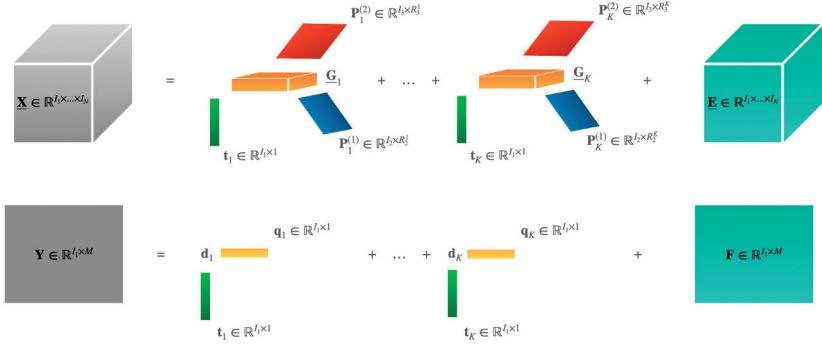


Figure 5.1: Scheme of eBTTR algorithm with 3rd-order predictor variable $\underline{\mathbf{X}}$ and 2nd-order response variable \mathbf{Y} . Note that each block is computed using ACE. In this work, $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples} \times \text{Channels} \times \text{Frequencies} \times \text{Time}}$ and $\mathbf{Y} \in \mathbb{R}^{\text{Samples} \times \text{Fingers}}$.

on (L_1^k, \dots, L_N^k) BTD with automatic MTR determination. Specifically, it is a deflation-based method that sequentially decomposes $\underline{\mathbf{X}}$ and \mathbf{Y} into a series of blocks of maximally correlated representations extracted via Automatic Component Extraction (ACE). A scheme of eBTTR is shown in Figure 5.1.

Given a set of training data $\underline{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and vectoral response $\mathbf{Y}_{\text{train}} \in \mathbb{R}^{I_1 \times M}$, eBTTR training consists of automatically identifying K blocks s.t.

$$\underline{\mathbf{X}}_{\text{train}} = \sum_{k=1}^K \underline{\mathbf{G}}_k \times_1 \mathbf{t}_k \times_2 \mathbf{P}_k^{(2)} \times_3 \dots \times_N \mathbf{P}_k^{(n)} + \underline{\mathbf{E}}_k$$

$$\mathbf{Y}_{\text{train}} = \sum_{k=1}^K \mathbf{u}_k \mathbf{q}_k^T + \mathbf{F}_k \text{ with } \mathbf{u}_k = \mathbf{t}_k b_k$$

with $\underline{\mathbf{G}}_k \in \mathbb{R}^{1 \times R_2^k \times \dots \times R_N^k}$ the core tensor for the k th-block, $\mathbf{P}_k^{(n)}$ the k th loading matrix for the n -mode, \mathbf{u}_k and \mathbf{t}_k latent components, \mathbf{q}_k the loading matrix, b_k the regression coefficient, and $\underline{\mathbf{E}}_k$ and \mathbf{F}_k residuals. Once the model is trained – and, hence, $\underline{\mathbf{G}}_k$, $\mathbf{P}_k^{(n)}$ and b_k are computed – the final

```

Input:  $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ ,  $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$ ,  $K$ 
Output:  $\{\mathbf{P}_k^{(n)}\}, \{\mathbf{t}_k\}, \{\mathbf{q}_k\}, \underline{\mathbf{G}}_k^{(X)}$  for  $k = 1, \dots, K$ ;  $n = 2, \dots, N$ 
1: Initialisation of  $\underline{\mathbf{E}}_1 = \underline{\mathbf{X}}$  and  $\mathbf{F}_1 = \mathbf{Y}$ 
2: for  $k = 1$  to  $K$  do
3:   if  $\|\underline{\mathbf{E}}_k\| > \epsilon$  and  $\|\mathbf{F}_k\| > \epsilon$  then
4:      $\underline{\mathbf{C}}_k = \langle \underline{\mathbf{E}}_k, \mathbf{F}_k \rangle_{\{1,1\}}$ 
5:      $\underline{\mathbf{G}}_k^{(X)}, \mathbf{q}_k, \mathbf{t}_k, \mathbf{P}_k^{(2)}, \dots, \mathbf{P}_k^{(N)} = ACE(\underline{\mathbf{E}}_k, \mathbf{F}_k)$ 
6:      $\mathbf{u}_k = \mathbf{F}_k \mathbf{q}_k$ 
7:      $\mathbf{d}_k = \mathbf{u}_k^T \mathbf{t}_k$ 
    {Deflation:}
8:      $\underline{\mathbf{E}}_{k+1} = \underline{\mathbf{E}}_k - [\underline{\mathbf{G}}_k^{(X)}; \mathbf{t}_k, \mathbf{P}_k^{(2)}, \dots, \mathbf{P}_k^{(N)}]$ 
9:      $\mathbf{F}_{k+1} = \mathbf{F}_k - \mathbf{d}_k \mathbf{t}_k \mathbf{q}_k^T$ 
10:   else
11:     break
12:   end if
13: end for

```

Figure 5.2: eBTTR

prediction is obtained as follows: $\mathbf{Y}_{\text{test}} = \mathbf{T}\mathbf{Z} = \underline{\mathbf{X}}_{\text{test}(1)}\mathbf{W}\mathbf{Z}$ where each column $\mathbf{w}_k = (\mathbf{P}_k^{(n)} \otimes \dots \otimes \mathbf{P}_k^{(2)})vec(\underline{\mathbf{G}}_k)$ and each row $z_k = b_k q_k$. This is summarized in Algorithm 5.2.

Automatic Component Extraction (ACE)

Given an N -way variable $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ and a vectorial variable $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$, we aim to automatically extract the latent components \mathbf{t} , \mathbf{q} and $P^{(n)}_{(n=2)}^N$, associated with the n -th mode of $\underline{\mathbf{X}}$ and maximally correlated with \mathbf{Y} , while $\left\| \underline{\mathbf{X}} - [\underline{\mathbf{G}}; \mathbf{t}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}] \right\|_F$ is minimized.

Within ACE, we define the mode-1 cross-product between predictor and response variables as $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$ and its decomposition as $\underline{\mathbf{C}} \approx [\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}]$. We provide the model with automatic SNR and τ selection based on Bayesian Information Criterion (BIC) defined here as:

$$BIC(\tau, \text{SNR} | \text{SNR}, \tau^*) =$$

$$\log\left(\frac{\left\| \underline{\mathbf{C}} - [\underline{\mathbf{G}}^{(c)}; \mathbf{q}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(N)}] \right\|_F}{s}\right) + \frac{\log(s)}{s} DF, \quad (5.1)$$

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$

Output: $\underline{\mathbf{G}}^{(X)} \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$, $\mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

- 1: $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{Y} \rangle_{(1)}$
- 2: **Initialisation** of $\tau = 90, \dots, 100$; $SNR = 1, \dots, 50$
- 3: **for** SNR_i in SNR **do**
- 4: **for** τ_j in τ **do**
- 5: $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N = mPSTD(\underline{\mathbf{X}}, \mathbf{Y}, SNR_i, \tau_j)$
- 6: calculate BIC value corresponding to SNR_i and τ_j using Eq 5.1
- 7: **end for**
- 8: select $\tau^* = \operatorname{argmin}_\tau \text{BIC}(\tau)$
- 9: calculate BIC value corresponding to SNR_i and τ^* using Eq 5.1
- 10: **end for**
- 11: select $SNR^* = \operatorname{argmin}_{SNR} \text{BIC}(SNR, \tau^*)$
- 12: $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N = mPSTD(\underline{\mathbf{X}}, \mathbf{Y}, SNR^*, \tau^*)$
- 13: $\mathbf{t} = (\underline{\mathbf{X}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(N)T})_{(1)} \operatorname{vec}(\underline{\mathbf{G}})$
- 14: $\mathbf{t} = \mathbf{t} / \|\mathbf{t}\|_F$
- 15: $\underline{\mathbf{G}}^{(X)} = [\underline{\mathbf{X}}; \mathbf{t}^T, \mathbf{P}^{(2)T}, \dots, \mathbf{P}^{(N)T}]$
- 16: **return** $\underline{\mathbf{G}}^{(X)}, \mathbf{q}, \mathbf{t}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Figure 5.3: ACE

where $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are the sparse core, latent vector and factor matrices obtained with mPSTD [28] - a modified version of the original sparse Tucker decomposition (PSTD) [31] - using specific τ and SNR values, s the number of entries in $\underline{\mathbf{G}}$, and DF the degree of freedom calculated as the number of non-zero elements in $\underline{\mathbf{G}}^{(c)}$, as suggested in [32]. For each SNR value, the associated optimal τ is computed as $\tau^* = \operatorname{argmin}_\tau \text{BIC}(\tau, \text{SNR})$. Then, the optimal SNR is determined as $SNR^* = \operatorname{argmin}_{SNR} \text{BIC}(SNR, \tau^*)$. Once $\underline{\mathbf{G}}^{(c)}$, q and $\{\mathbf{P}^{(n)}\}_{n=2}^N$ are computed, the score vector t is first calculated as

$$\mathbf{t} = (\underline{\mathbf{C}} \times_2 \mathbf{P}^{(2)T} \times_3 \dots \times_N \mathbf{P}^{(N)T})_{(1)} \operatorname{vec}(\underline{\mathbf{G}}^{(c)}),$$

and then normalized. This is summarized in Algorithm 5.3.

The mPSTD model is first initialized with higher-order orthogonal iteration (HOOI) [33]. Then, iteratively, a soft-thresholding rule based on parameter λ , alternated with a threshold τ , are applied to enhance model sparsity and to prune irrelevant components, respectively. Note that in [31] $\text{SNR} \in [1, 50]$ is used to derive, via a line search, the optimal degree of sparsity λ of the core tensor (see [31]). At each iteration, the core tensor $\underline{\mathbf{G}}$ is updated using the soft-thresholding rule as $\underline{\mathbf{G}} = \operatorname{sgn}(\underline{\mathbf{G}}) \times \max\{|\underline{\mathbf{G}}| - \lambda, 0\}$, while the threshold $\tau \in [0, 100]$ is

Input: $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $\mathbf{Y} \in \mathbb{R}^{I_1 \times M}$, τ, SNR
Output: $\underline{\mathbf{G}} \in \mathbb{R}^{1 \times R_2 \times \dots \times R_N}$, $\mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Initialisation :

- 1: $\underline{\mathbf{C}} = \langle \underline{\mathbf{X}}, \mathbf{y} \rangle_{(1)} \in \mathbb{R}^{1 \times I_2 \times \dots \times I_N}$
- 2: **Initialisation of** $\{\mathbf{P}^{(n)}\}_{n=2}^N$, \mathbf{q} and $\underline{\mathbf{G}}$ using HOOI on $\underline{\mathbf{C}}$
LOOP Process
- 3: **repeat**
- 4: **update** $\underline{\mathbf{G}}$ using SNR
- 5: **prune** $\{\overline{\mathbf{P}}^{(n)}\}_{n=2}^N$, \mathbf{q} and $\underline{\mathbf{G}}$ using τ
- 6: **until** convergence is reached
- 7: **return** $\underline{\mathbf{G}}, \mathbf{q}, \{\mathbf{P}^{(n)}\}_{n=2}^N$

Figure 5.4: mPSTD

used to reject unnecessary components from the n-mode $S^{(n)} = \{r|100(1 - \frac{\sum_i \mathbf{G}_{(n)(r,i)}}{\sum_{t,i} \mathbf{G}_{(n)(t,i)}}) \geq \tau\}$, $\mathbf{P}^{(n)} = \mathbf{P}^{(n)}(:, S^{(n)})$, $\mathbf{q} = \mathbf{q}(S^{(n)})$ and $\mathbf{G}^{(n)} = \mathbf{G}^{(n)}(S^{(n)}, :)$.

The mPSTD is summarized in Algorithm 5.4.

5.3.2 Non-multilinear approaches

We will consider the BCI competition IV dataset (see further) for comparing the performance of eBTTR with the winner of this competition, a linear regression model based on amplitude modulation (AM) [17], as well as with more recent attempts based on Random Forests (RF), Convolutional Neural Networks (CNN), and Long Short-Term Memory Network (LSTM) (proposed and compared in [18]), all of which are non-multilinear approaches.

Since in both [17] and [18] only 1 test set was used to assess performance, which factually impedes statistically testing the significance of any observed performance difference between the compared algorithms, we have re-implemented AM, RF, LARS, CNN and LSTM. We proceeded as follows.

For AM, we replicated the procedure described in [17]. First, we filtered the ECoG signals in the sub-gamma (1-60Hz), gamma (60-100 Hz) and high-gamma bands (100-200 Hz). For each of these bands, we determined the amplitude modulation and used it to estimate their band-specific AM features. For each finger and subject, we used forward feature selection using a wrapper approach to find the relevant AM features. These features are then used in a linear regression model. Finally, we verified whether the obtained performance compared with the one reported in [17].

For LARS, we used the *LassoLars* function of the most recent version of the *scikit-learn* package of Python (version 0.24.2 released in April 2021; we assume that Xie et al. used version 0.19.1, but this was not reported). LARS transforms the original signal using ICA, decomposes it into different bands, calculates band powers and fits a LassoLars model. LassoLars has one main parameter, α , the multiplier for the penalty term whereby $\alpha = 0$ corresponds to ordinary least square linear regression. A line search was used to optimize α .

For RF, a similar processing pipeline was used. The *RandomForestRegressor* function of the *scikit-learn* package was used (same version as above). RF transforms the original signal using ICA, decomposes it into different bands, calculates band powers and fits the *RandomForestRegressor* model.

The CNN and LSTM were built according to the specifications and architecture given in [18]. CNN refers to a linear regression model applied to features prior extracted using a CNN. The CNN and LSTM were build using PyTorch in Python (version 1.8.0 released in March 2021; note that Xie et al. did not specify which version they used).

5.3.3 Dataset

We will compare eBTTR’s performance with that of the aforementioned non-multilinear models, as well as with multilinear models HOPLS and BTTR, in predicting continuous finger flexions from ECoG recordings. We will use hereto the publically-available BCI Competition IV dataset 4 for reproducibility’s sake. It comprises ECoG signals sampled at 1000 Hz from the motor cortex (hand-knob area) of three subjects, as well as the time courses of the flexion of each of five fingers of the contralateral hand (gauged with a data glove). There are 150 trials (samples) in total (30 trials per finger) recorded in a single session (600 seconds). Subjects flexed the cued finger 3-5 times for 2 seconds followed by a rest period of 2 seconds. The first 400 and the last 200 seconds of recording were used as training and testing sets, respectively. The number of ECoG electrodes (channels) was 62 for Subject 1, 48 for Subject 2, and 64 for Subject 3. More details about the data can be found in [34].

The dataglove position measurement lags by 37ms ($\pm 3\text{ms}$, SEM) the amplifier measurement. However, this is of the same order of granularity as the datagloves position measurement as it is sampled at 25Hz, thus every 40ms. Hence, the dataglove position measurement is shifted by 1 position in order to account for the lag. This is also done in the approaches against which we compare our (e)BTTR’s performance.

The glove data and the ECoG signals were extracted starting 1 sec prior to trial onset (“epoch”). ECoG recordings were prepared first by filtering out the power line using notch filters centered at 50 and 100 Hz, then, by removing bad

channels (i.e., those exhibiting unstable or unchanging signals, i.e. channels 55 in subject 1, channels 21 and 38 in subject 2, and channel 50 in subject 3). The remaining channels were re-referenced to a Common Average Reference (CAR) [3]: the average of all signals is taken as reference and subtracted from all signals [35]. The ECoG signals are transformed into 4-th order ECoG tensor $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples} \times \text{Channels} \times \text{Frequencies} \times \text{Time}}$ as follows:

- **Samples** depends on the length of the data glove's trajectory vector Y .
- **Channels** corresponds to the number of curated electrodes (thus, after removing bad ones), i.e., 61 for Subject 1, 46 for Subject 2, and 63 for Subject 3.
- **Frequencies** corresponds to the 8 bidirectional fourth-order Butterworth band-pass filters [36] ECoG signals are subjected to extract the corresponding spectral amplitudes in the δ (1.5 - 5 Hz), θ (5 - 8 Hz), α (8 - 12 Hz), β_1 (12 - 24 Hz), β_2 (24 - 34 Hz), γ_1 (34 - 60 Hz), γ_2 (60 - 100 Hz), γ_3 (100 - 130 Hz) bands.
- **Time** is composed of 10 instances or bins as for each of the described 8 components, the most recent 1 second epoch is downsampled to 10 Hz.

The data glove data is normalized (z-scored) independently for each finger, yielding a vector $Y \in \mathbb{R}^{\text{Samples} \times \text{Fingers}}$. **Samples** corresponds to the sampled finger flexions over time. The **Fingers** dimension corresponds to the 5 fingers, Thumb, Index, Middle, Ring and Pinky.

As a result, for each channel c , a 3rd order ECoG tensor $\underline{\mathbf{X}}_c \in \mathbb{R}^{\text{Samples} \times \text{Frequencies} \times \text{Time}}$ is computed after epoch selection, band-pass filtering and 10 Hz downsampling. These steps are repeated for each time sample s . The results are then merged into a matrix $\mathbf{X}_{c,f} \in \mathbb{R}^{\text{Samples} \times \text{Time (10 bins)}}$. This matrix is then normalized (z-scored) to reduce the difference in magnitude between frequency bands: $x_{s,t} = (x_{s,t} - \mu_{c,f})/\delta_{c,f}$ where $\mu_{c,f}$ and $\delta_{c,f}$ are, respectively, the mean and standard deviation computed for channel c and frequency band f of the training set. Note that the same values are used to normalize the test set.

Once all $\mathbf{X}_{c,f} \in \mathbb{R}^{\text{Samples} \times \text{Time (10)}}$ are computed for the various components (8), they are merged into $\underline{\mathbf{X}}_c \in \mathbb{R}^{\text{Samples} \times \text{Frequencies (8)} \times \text{Time (10)}}$.

Next, when all 3rd order ECoG tensors $\underline{\mathbf{X}}_c \in \mathbb{R}^{\text{Samples} \times \text{Frequencies (8 components)} \times \text{Time (10 bins)}}$ are computed for each channel c , they are merged into $\underline{\mathbf{X}} \in \mathbb{R}^{\text{Samples} \times \text{Channels } \{61, 46, 63\} \times \text{Frequen}}$

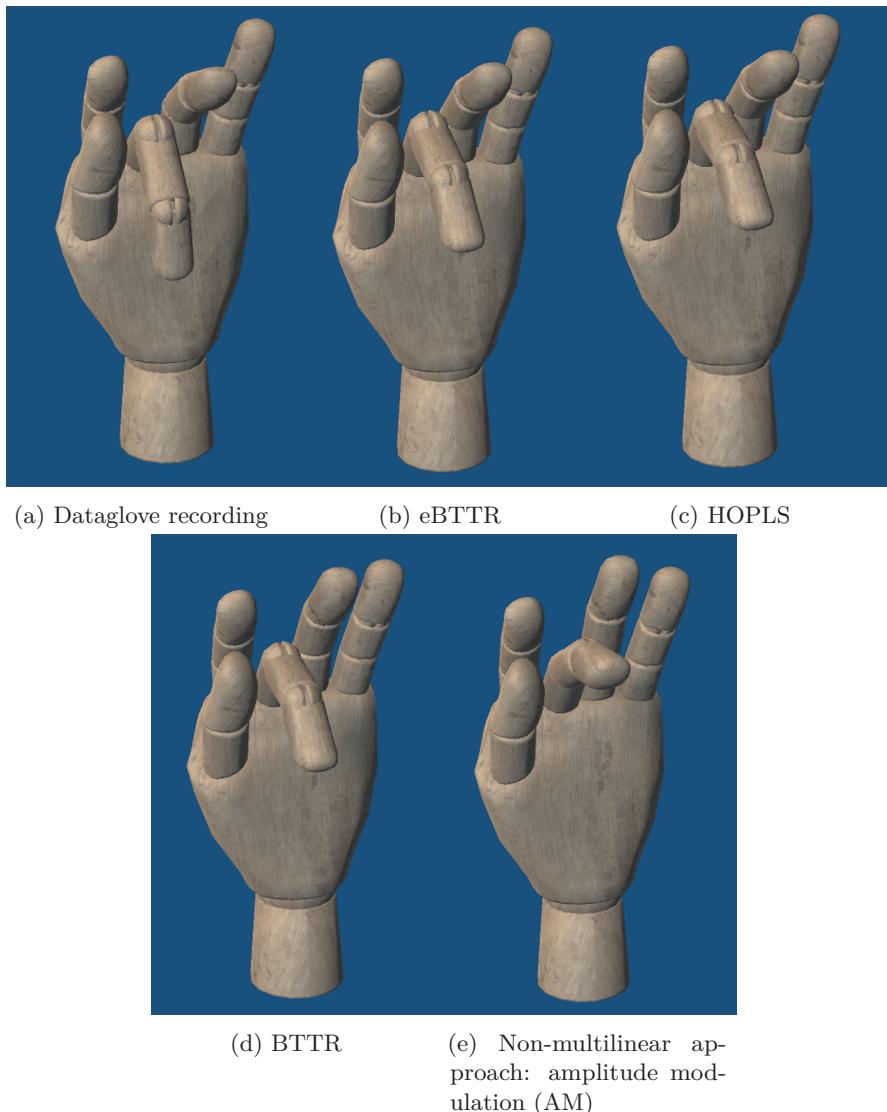


Figure 5.5: Unintentional coactivation for subject 3 during middle finger movement (visualized as left hand).

5.3.4 Parameter optimization multilinear models, performance assessment

In order to optimize the model parameters from the training data, K, R_2, \dots, R_N for HOPLS and K for BTTR and eBTTR, a 5-fold cross-validation approach was used.

Since the cited BCI Competition IV dataset 4 studies reported Pearson's correlation coefficients between data glove- and predicted intended finger trajectories, we also applied it here. In support of the statistical analysis, the test data was split in 5 non-overlapping blocks. We used the two-tailed Wilcoxon signed-rank test [37] to compare average accuracies per finger and subject; they are considered significantly different if the p-value is < 0.05 . The non-intended finger movements (coactivations) are visually inspected from hand avatar movies.

5.4 Results

The Pearson correlation coefficients are reported for the intended movements of each finger individually, and averaged across all fingers except for finger 4 (ring) as flexing the latter is difficult to suppress when the 3rd or 5th finger is flexing, as this was done by the authors of the cited papers (note that we assess coactivations qualitatively, see further). The average results and their standard deviations for the 5 blocks of test data (see above) are listed in Table 5.2, Table 5.3 and Table 5.4 (listed under Avg.) for subjects 1, 2 and 3, respectively.

A statistically significant difference between BTTR and eBTTR was found for the middle finger of subject 1 and subject 2 and between eBTTR and HOPLS for subjects 1 and 2, mainly due to the difference in correlation coefficients of the middle finger.

We also investigated the runtime needed to estimate the model parameters. One of the main advantages of BTTR is its speed. BTTR is trained significantly faster than HOPLS (e.g. 3 minutes against 14 hours for HOPLS). Note that HOPLS requires computationally expensive techniques such as cross-validation to identify the optimal set of model parameters (i.e., the number of scores and loadings). Since eBTTR is based on BTTR, it enjoys the latter's fast training time (e.g. 5 minutes against 3 minutes for BTTR).

Figure 5.5 shows stills of the hand avatar replicating the predicted movements of a few models. Compared to BTTR and AM, we observe that eBTTR and HOPLS better decode joint finger movements, comprising intentional- (cued

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
eBTTR	$0.71 \pm .04$	$0.75 \pm .06$	$0.49 \pm .06$	$0.69 \pm .04$	$0.68 \pm .01$	$0.66 \pm .03$
BTTR	$0.72 \pm .06$	$0.77 \pm .09$	$0.38 \pm .02$	$0.68 \pm .04$	$0.67 \pm .02$	$0.64 \pm .05$
HOPLS	$0.70 \pm .05$	$0.79 \pm .08$	$0.36 \pm .03$	$0.70 \pm .06$	$0.65 \pm .02$	$0.63 \pm .04$
AM	$0.57 \pm .03$	$0.69 \pm .06$	$0.14 \pm .02$	$0.52 \pm .04$	$0.28 \pm .01$	$0.42 \pm .03$
RF	$0.58 \pm .09$	$0.54 \pm .05$	$0.07 \pm .03$	$0.31 \pm .05$	$0.33 \pm .02$	$0.38 \pm .05$
LARS	$0.11 \pm .05$	$0.08 \pm .03$	$0.10 \pm .02$	$0.60 \pm .05$	$0.39 \pm .02$	$0.17 \pm .03$
CNN	$0.67 \pm .04$	$0.78 \pm .04$	$0.11 \pm .02$	$0.54 \pm .03$	$0.45 \pm .04$	$0.50 \pm .04$
LSTM	$0.73 \pm .03$	$0.79 \pm .08$	$0.18 \pm .02$	$0.61 \pm .04$	$0.45 \pm .04$	$0.54 \pm .04$

Table 5.2: Intended (cued) finger movement accuracy for Subject 1 (Pearson correlation).

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
eBTTR	$0.63 \pm .05$	$0.47 \pm .08$	$0.33 \pm .03$	$0.52 \pm .02$	$0.47 \pm .01$	$0.48 \pm .05$
BTTR	$0.64 \pm .05$	$0.46 \pm .08$	$0.27 \pm .04$	$0.50 \pm .03$	$0.48 \pm .01$	$0.46 \pm .05$
HOPLS	$0.63 \pm .04$	$0.47 \pm .06$	$0.26 \pm .05$	$0.51 \pm .02$	$0.48 \pm .01$	$0.44 \pm .04$
AM	$0.52 \pm .03$	$0.36 \pm .06$	$0.23 \pm .02$	$0.48 \pm .04$	$0.33 \pm .01$	$0.36 \pm .03$
RF	$0.52 \pm .05$	$0.36 \pm .04$	$0.22 \pm .03$	$0.39 \pm .04$	$0.25 \pm .02$	$0.34 \pm .04$
LARS	$0.54 \pm .05$	$0.41 \pm .04$	$0.18 \pm .02$	$0.44 \pm .04$	$0.25 \pm .02$	$0.35 \pm .03$
CNN	$0.60 \pm .04$	$0.40 \pm .04$	$0.24 \pm .02$	$0.44 \pm .03$	$0.28 \pm .04$	$0.38 \pm .04$
LSTM	$0.62 \pm .03$	$0.38 \pm .08$	$0.27 \pm .02$	$0.47 \pm .04$	$0.30 \pm .04$	$0.39 \pm .04$

Table 5.3: Idem to Table 5.2 but for Subject 2.

finger) and unintentional ones (coactivations).¹ The latter are in part due to the subject’s hand- and forearm tendon anatomy and the divergent connections from hand cortical motor neurons to spinal motor neurons, limiting cortical access to individual fingers [38]. Co-activations are therefore present in the data glove data, but most likely not encoded by the ECoG signals. One could question whether these coactivations should be decoded at all. However, rendering them makes the outcome look more natural and could provide additional sensory information when approaching a physical object.

5.5 Conclusion

Tensor-based decoders are able to exploit multiway structured data better than those that rely on data unfolding. Decoding joint finger movements is a challenging task that needs to be addressed when envisaging BCI-based hand prosthetic- or exoskeleton control in realistic object handling scenarios. To tackle this, we proposed an extension of the tensor decomposition approach we developed before so that it can be used to regress over multiple fingers

¹For a quantitative comparison in the case of eBTTR and BTTR, we refer to the Supplementary Material section.

Methods	Thumb	Index	Middle	Ring	Pinky	Avg.
eBTTR	0.71 ± .05	0.57 ± .07	0.64 ± .04	0.62 ± .02	0.73 ± .01	0.66 ± .05
BTTR	0.73 ± .05	0.59 ± .08	0.64 ± .04	0.63 ± .02	0.72 ± .01	0.67 ± .05
HOPLS	0.74 ± .06	0.57 ± .09	0.65 ± .02	0.61 ± .04	0.68 ± .02	0.64 ± .04
AM	0.59 ± .03	0.51 ± .06	0.32 ± .02	0.53 ± .04	0.42 ± .01	0.46 ± .03
RF	0.67 ± .05	0.27 ± .04	0.16 ± .03	0.14 ± .04	0.36 ± .02	0.37 ± .04
LARS	0.72 ± .05	0.43 ± .04	0.45 ± .02	0.51 ± .04	0.64 ± .02	0.56 ± .03
CNN	0.74 ± .03	0.53 ± .05	0.45 ± .04	0.49 ± .03	0.68 ± .06	0.60 ± .05
LSTM	0.74 ± .02	0.55 ± .06	0.46 ± .04	0.41 ± .02	0.75 ± .06	0.62 ± .05

Table 5.4: Idem to Table 5.2 but for Subject 3.

simultaneously and showed that it can challenge state-of-the-art multiway regression techniques while outperforming more conventional ones.

5.6 Funding

AF was supported by the Belgian Fund for Scientific Research – Flanders (FWO 1157019N). MMVH is supported by research grants received from the European Union’s Horizon 2020 research and innovation programme onder grant agreement No. 857375, the special research fund of the KU Leuven (C24/18/098), the Belgian Fund for Scientific Research – Flanders (G0A4118N, G0A4321N, G0C1522N), and the Hercules Foundation (AKUL 043).

Bibliography

- [1] *Who spinal cord injury*, <https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury>, Accessed: 2022-06-30.
- [2] A. B. Ajiboye, F. R. Willett, D. R. Young, W. D. Memberg, B. A. Murphy, J. P. Miller, B. L. Walter, J. A. Sweet, H. A. Hoyen, M. W. Keith, *et al.*, “Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration”, *The Lancet*, vol. 389, no. 10081, pp. 1821–1830, 2017.
- [3] C. E. Bouton, A. Shaikhouni, N. V. Annetta, M. A. Bockbrader, D. A. Friedenberg, D. M. Nielson, G. Sharma, P. B. Sederberg, B. C. Glenn, W. J. Mysiw, *et al.*, “Restoring cortical control of functional movement in a human with quadriplegia”, *Nature*, vol. 533, no. 7602, pp. 247–250, 2016.
- [4] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia”, *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.
- [5] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, “High-performance neuroprosthetic control by an individual with tetraplegia”, *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.
- [6] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simner, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt, *et al.*, “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm”, *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.
- [7] B. Wodlinger, J. Downey, E. Tyler-Kabara, A. Schwartz, M. Boninger, and J. Collinger, “Ten-dimensional anthropomorphic arm control in a human brain-machine interface: Difficulties, solutions, and limitations”, *Journal of neural engineering*, vol. 12, no. 1, p. 016011, 2014.

- [8] K. J. Miller, L. B. Sorensen, J. G. Ojemann, and M. Den Nijs, “Power-law scaling in the brain surface electric potential”, *PLoS Comput Biol*, vol. 5, no. 12, e1000609, 2009.
- [9] R. J. Staba, C. L. Wilson, A. Bragin, I. Fried, and J. Engel Jr, “Quantitative analysis of high-frequency oscillations (80–500 hz) recorded in human epileptic hippocampus and entorhinal cortex”, *Journal of neurophysiology*, vol. 88, no. 4, pp. 1743–1752, 2002.
- [10] W. Wang, J. L. Collinger, A. D. Degenhart, E. C. Tyler-Kabara, A. B. Schwartz, D. W. Moran, D. J. Weber, B. Wodlinger, R. K. Vinjamuri, R. C. Ashmore, *et al.*, “An electrocorticographic brain interface in an individual with tetraplegia”, *PloS one*, vol. 8, no. 2, e55344, 2013.
- [11] E. S. Nurse, S. E. John, D. R. Freestone, T. J. Oxley, H. Ung, S. F. Berkovic, T. J. O’Brien, M. J. Cook, and D. B. Grayden, “Consistency of long-term subdural electrocorticography in humans”, *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 2, pp. 344–352, 2017.
- [12] M. J. Vansteensel, E. G. Pels, M. G. Bleichner, M. P. Branco, T. Denison, Z. V. Freudenburg, P. Gosselaar, S. Leinders, T. H. Ottens, M. A. Van Den Boom, *et al.*, “Fully implanted brain–computer interface in a locked-in patient with als”, *New England Journal of Medicine*, vol. 375, no. 21, pp. 2060–2066, 2016.
- [13] A. L. Benabid, T. Costecalde, A. Eliseyev, G. Charvet, A. Verney, S. Karakas, M. Foerster, A. Lambert, B. Morinière, N. Abroug, *et al.*, “An exoskeleton controlled by an epidural wireless brain–machine interface in a tetraplegic patient: A proof-of-concept demonstration”, *The Lancet Neurology*, vol. 18, no. 12, pp. 1112–1122, 2019.
- [14] J. Kubanek, K. J. Miller, J. G. Ojemann, J. R. Wolpaw, and G. Schalk, “Decoding flexion of individual fingers using electrocorticographic signals in humans”, *Journal of neural engineering*, vol. 6, no. 6, p. 066001, 2009.
- [15] Z. Wang, Q. Ji, K. J. Miller, and G. Schalk, “Decoding finger flexion from electrocorticographic signals using a sparse gaussian process”, in *2010 20th International conference on pattern recognition*, IEEE, 2010, pp. 3756–3759.
- [16] R. Flamary and A. Rakotomamonjy, “Decoding finger movements from ecog signals using switching linear models”, *Frontiers in neuroscience*, vol. 6, p. 29, 2012.
- [17] N. Liang and L. Bougrain, “Decoding finger flexion from band-specific ecog signals in humans”, *Frontiers in neuroscience*, vol. 6, p. 91, 2012.
- [18] Z. Xie, O. Schwartz, and A. Prasad, “Decoding of finger trajectory from ecog using deep learning”, *Journal of neural engineering*, vol. 15, no. 3, p. 036009, 2018.
- [19] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications:

- From two-way to multiway component analysis”, *IEEE signal processing magazine*, vol. 32, no. 2, pp. 145–163, 2015.
- [20] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications”, *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
 - [21] L. De Lathauwer, “Decompositions of a higher-order tensor in block terms—part i: Lemmas for partitioned matrices”, *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1022–1032, 2008.
 - [22] ——, “Decompositions of a higher-order tensor in block terms—part ii: Definitions and uniqueness”, *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.
 - [23] L. De Lathauwer and D. Nion, “Decompositions of a higher-order tensor in block terms—part iii: Alternating least squares algorithms”, *SIAM journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1067–1083, 2008.
 - [24] B. Hunyadi, D. Camps, L. Sorber, W. Van Paesschen, M. De Vos, S. Van Huffel, and L. De Lathauwer, “Block term decomposition for modelling epileptic seizures”, *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–19, 2014.
 - [25] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. C. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, “Higher order partial least squares (hopls): A generalized multilinear regression method”, *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1660–1673, 2012.
 - [26] A. Eliseyev and T. Aksanova, “Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording”, *PloS one*, vol. 11, no. 5, e0154878, 2016.
 - [27] F. Camarrone, M. P. Branco, N. F. Ramsey, and M. M. Van Hulle, “Accurate offline asynchronous detection of individual finger movement from intracranial brain signals using a novel multiway approach”, *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 7, pp. 2176–2187, 2020.
 - [28] A. Faes, F. Camarrone, and M. M. Van Hulle, “Single finger trajectory prediction from intracranial brain activity using block-term tensor regression with fast and automatic component extraction”, *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
 - [29] K. J. Miller, C. J. Honey, D. Hermes, R. P. Rao, J. G. Ojemann, *et al.*, “Broadband changes in the cortical surface potential track activation of functionally diverse neuronal populations”, *Neuroimage*, vol. 85, pp. 711–720, 2014.
 - [30] G. Pan, J.-J. Li, Y. Qi, H. Yu, J.-M. Zhu, X.-X. Zheng, Y.-M. Wang, and S.-M. Zhang, “Rapid decoding of hand gestures in electrocorticography using recurrent neural networks”, *Frontiers in neuroscience*, vol. 12, p. 555, 2018.

- [31] T. Yokota and A. Cichocki, “Multilinear tensor rank estimation via sparse tucker decomposition”, in *2014 Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS)*, IEEE, 2014, pp. 478–483.
- [32] G. I. Allen and M. Maletić-Savatić, “Sparse non-negative generalized pca with applications to metabolomics”, *Bioinformatics*, vol. 27, no. 21, pp. 3029–3035, 2011.
- [33] L. De Lathauwer, B. De Moor, and J. Vandewalle, “On the best rank-1 and rank-(r₁, r₂,..., r_n) approximation of higher-order tensors”, *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.
- [34] K. J. Miller and G. Schalk, “Prediction of finger flexion: 4th brain-computer interface data competition”, *BCI Competition IV*, vol. 1, pp. 1–2, 2008.
- [35] C. D. Binnie, *Clinical neurophysiology: EEG, paediatric neurophysiology, special techniques and applications*. Elsevier Health Sciences, 2003, vol. 2.
- [36] L. R. Tucker, “Some mathematical notes on three-mode factor analysis”, *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [37] F. Wilcoxon, “Individual comparisons by ranking methods”, in *Breakthroughs in statistics*, Springer, 1992, pp. 196–202.
- [38] C. E. Lang and M. H. Schieber, “Human finger independence: Limitations due to passive mechanical coupling versus active neuromuscular control”, *Journal of neurophysiology*, vol. 92, no. 5, pp. 2802–2810, 2004.

Chapter 6

Conclusion

6.1 General Conclusion

The objective of this PhD was to investigate and model motor-related ECoG- and scalp EEG recordings. First, we performed source localization and extracted connectivity patterns from motor-related EEG signals. This led to Sparse eLORETA, a novel method for estimating a nonparametric solution to the source localization problem. Its goal is to generate a sparser solution compared to other source localization methods, including eLORETA, while benefitting from the latter's superior source localization accuracy (Figure 6.1). Regarding connectivity extraction, we found that some types of neural networks, in particular LSTMs, may be considered for estimating the directed connectivity of reconstructed EEG sources. However, no method is flawless, and we showed that even an established method such as TRGC can generate faulty estimates, which is something future research could built upon. There is much potential for a hybrid approach, in which a neural network is used as a preprocessing step to chart the interesting directed connectivity patterns, after which a conventional method is applied for the directed connectivities.

Second, regarding the objective to investigate motor-related ECoG, We wanted more concretely to demonstrate the feasibility of a multiway BCI system, based on human ECoG recordings, to decode the continuous time course of actual flexion/extension of individual fingers. we proposed a multiway Partial Least Squares framework for the accurate decoding of continuous finger trajectories while allowing for fast decoder (re-)calibration. Multi-way data processing is not new and most methods were created decades ago, but it is only since recently it has been deployed in various applications. Multiway methods are

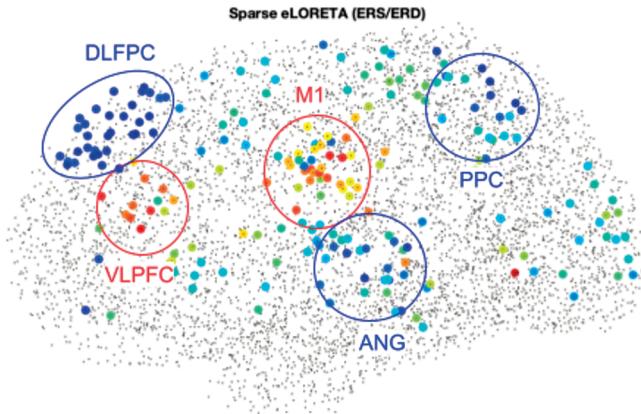


Figure 6.1: Contralateral view of the Sparse eLORETA solution with clusters labeled by the brain regions active during a motor task.

powerful as they respect the original data structure and detect hidden patterns across modalities. However, they are not widely used in BCI due to their high time-complexity. Indeed, while they can provide accurate predictions, the decoding process is often not fast enough to deliver real-time responses. Decoding processes can be slow due to (1) choosing the right parameters for the decoder and (2) creating and predicting multiway ECoG samples from continuous streams of data takes a lot of time. We started with Block-Term Tensor Regression (BTTR) [1], [2] which, similar to HOPLS, relies on a partial least squares approach (PLS) but differs conceptually by the decomposition of the mode-1 cross-product between predictor- and response variables, the blocks that can have different multilinear ranks (MTRs), and the use of the Automatic Component Extraction (ACE) for parameter estimation. These differences cause BTTR to be much faster to train than HOPLS with comparable accuracies. However, unlike BTTR, which can only predict a scalar value, such as the flexion of an individual finger, HOPLS is able to predict a vector, which renders the model –in principle– capable of predicting coordinated finger flexions, e.g. when grasping a cup. We then introduced extended Block-Term Tensor Regression (eBTTR) [3] in order to allow BTTR to also predict a vector, thus enabling the model capable of predicting coordinated finger flexions thereby also capturing finger co-activations, i.e. unintentional finger movements besides the one that is intentionally moved. These co-activations are likely not encoded by the ECoG signal, but decoded by the eBTTR model as it is trained to produce finger movements as they are recorded with a data glove (Fig. 6.5, left panel). The

eBTTR model thus forms a solid basis to decode coordinated finger movements and, in particular, for the sign language alphabet from brain recordings.

Block-Term Tensor Regression could potentially be applied to a multitude of multiway problems as they occur in sociology, climatology, process analysis, chemistry and computer vision [4]. For instance, the cokriging in sociology or climatology [5]. In computer vision, human motion reconstruction or tracking based on video sequences can also be regarded as multiway regression problems [5].



Figure 6.2: Left Panel: Digital data glove. Right Panel: Hand exoskeleton.

6.2 Future Research

6.2.1 Sign Language Alphabet Decoding

Brain computer interfaces (BCIs) that rely on imagined muscular activity to operate a text spelling device fail to provide communication beyond 10 words/minute [6]–[9], a rate much lower than the 150 words/min on average of natural speech. Even recent advances in speech BCIs, whose aim is to translate speech brain activity into text, albeit impressive, struggle to transmit more than 12 words/min even with a language model in place [10]. Inspired by the imagined handwriting decoding study of [11], and the advances in this doctorate in finger decoding from ECoG [2], [3], future research could bridge the communication gap by decoding sign language from imagined finger movements directly. A fresh look at text spelling was adopted by [11] when decoding imagined handwriting from hand motor brain activity. However, even though text spelling systems can enhance a paralyzed individual’s quality life, most users struggle to transmit more than 10 words/min, a rate much slower than that of natural verbal communication. Perhaps we can learn from deaf and hard-of-hearing people

that rely on sign language (Fig. 6.3) for their communication often on par with its verbal counterpart? When considered as a motor BCI solution for paralyzed individuals, this implies that we would be able to accurately decode imagined coordinated finger movements from brain recordings.

In order to show that this is in principle feasible, we conducted a pilot study (Fig. 6.4) using a dataset used before to classify 4 gestures from the American American Sign Language Alphabet [12]. Five patients (mean age 31, range 19-45) with intractable epilepsy were implanted with subdural ECoG grids covering the somatosensory hand motor area to localize their epileptogenic zone. The task, as described in ([13]), involved the execution of four different hand gestures from the American Sign Language finger spelling alphabet ('D', 'F', 'V' and 'Y') while jointly recording their ECoGs and data glove signals. The accuracy with which the trajectories were predicted (using eBTTR) looked promising (assessed in terms of the Pearson correlation). A further classification of these finger movement trajectories led to an improved gesture classification accuracy compared to that of the cited study. As sign language can be as efficient as speech [14], when successful, it could achieve faster communication rates than with current BCIs and, in this way, contribute to the development of improved communication solutions for paralyzed individuals.

6.2.2 Model training based on observed movement

In the case of an individual suffers from hand paralysis, we should decode letters of the sign language alphabet (Fig. 6.3) from ECoG activity elicited during finger motor imagery (imagined movement, IM). Decoder training is done, as usual, by recording motor cortical activity in response to actual (i.e. performed) movement (AM) [15]. However, since paralyzed individuals cannot engage or are hampered in intentional muscular activity, alternative approaches need to be developed. These can be based either on motor cortical activity associated with action observation (observed movement, OM) [16], in which case one person observes another performing an action, or on IM [17] in which case the subject imagines performing an action. In order to keep track of brain plasticity effects [18], as the user is gaining muscular control with the motor BCI in place, the latter also needs to be updated regularly.

6.2.3 Subject Training

Despite the interest it enjoys in the BCI community, motor imagery is a skill that needs to be learned, to select the appropriate control strategy [19], but a natural training procedure is still lacking. Current procedures rely on operant conditioning, simple visual feedback, and mental remapping of the actual motor task leading to long training periods, and even though tactile and proprioceptive

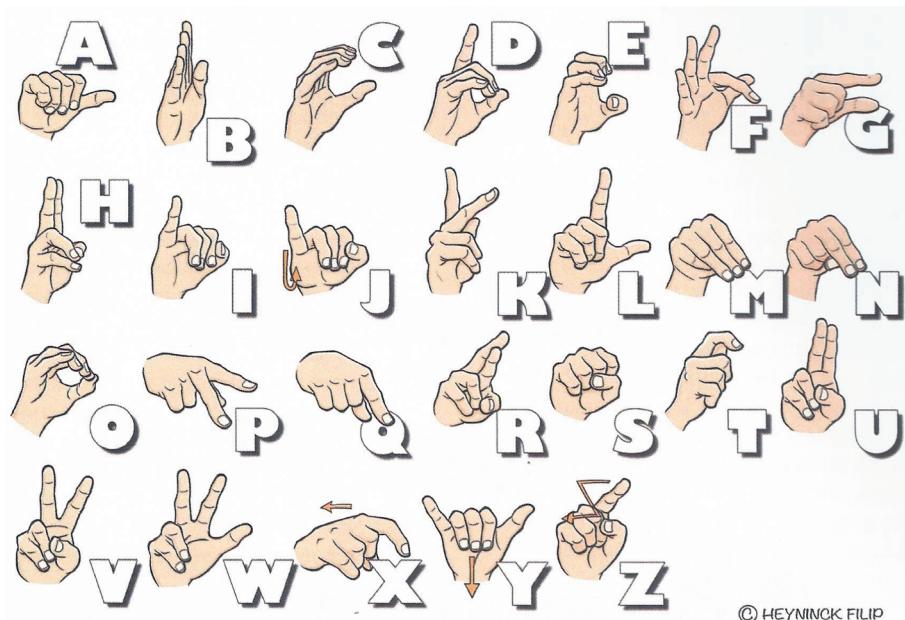


Figure 6.3: Excerpt of the Flemish sign language alphabet. An avatar is used to instruct the subject with the finger movements that lead to these signs, starting from a closed fist, to show the user what is decoded from his/her brain activity in real-time, or to have the attempts post-hoc evaluated by a sign language expert.

information is important to complete an action, it has often been neglected. With a hand exoskeleton, such as Gloreha Sinfonia's (see Figure 6.5 (Right Panel)) which does not obstruct the hand, palm or finger tips, subjects could rely on joint visual, tactile and proprioceptive feedback when performing a character from the sign language alphabet. Note that, although their proprioception could degrade, visual and tactile feedback are spared in ALS patients [20], at least over a long period. In addition, a recent BCI study on multi-tasking [21] showed that performance with tactile feedback was higher than with visual feedback. Finally, as prolonged subject training could result in brain plasticity changes, the characteristics of the ECoG signal could change, calling for occasional decoder retraining to maintain performance (recalibration).

6.2.4 Finger Abduction Decoding

As performing gestures from the sign language alphabet requires both finger flexion and finger abduction, both would need to be decoded. However,

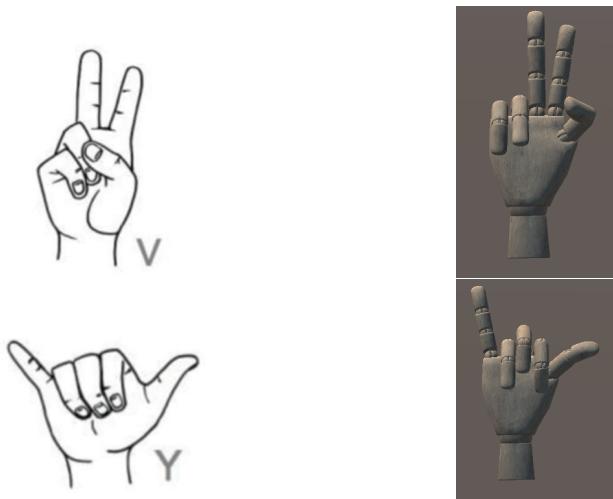


Figure 6.4: Outcome of our ECoG pilot study on performed sign language alphabet decoding using eBTTR. Left panel: the cued American sign language letters V and Y. Right panel: still images of the avatar of the decoded finger trajectories. The average Pearson correlations between the data glove- and decoded trajectories were above 70%.



Figure 6.5: Left Panel: Digital data glove. Right Panel: Hand exoskeleton.

unlike finger flexion, finger abduction decoding from ECoG has not yet been attempted.

Bibliography

- [1] F. Camarrone, M. P. Branco, N. F. Ramsey, and M. M. Van Hulle, “Accurate offline asynchronous detection of individual finger movement from intracranial brain signals using a novel multiway approach”, *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 7, pp. 2176–2187, 2020.
- [2] A. Faes, F. Camarrone, and M. M. Van Hulle, “Single finger trajectory prediction from intracranial brain activity using block-term tensor regression with fast and automatic component extraction”, *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [3] A. Faes and M. Van Hulle, “Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression”, *Journal of Neural Engineering*, 2022.
- [4] E. Acar and B. Yener, “Unsupervised multiway data analysis: A literature survey”, *IEEE transactions on knowledge and data engineering*, vol. 21, no. 1, pp. 6–20, 2008.
- [5] J. Liu, C. Zhu, Z. Long, Y. Liu, *et al.*, “Tensor regression”, *Foundations and Trends® in Machine Learning*, vol. 14, no. 4, pp. 379–565, 2021.
- [6] V. Gilja, C. Pandarinath, C. H. Blabe, P. Nuyujukian, J. D. Simeral, A. A. Sarma, B. L. Sorice, J. A. Perge, B. Jarosiewicz, L. R. Hochberg, *et al.*, “Clinical translation of a high-performance neural prosthesis”, *Nature medicine*, vol. 21, no. 10, pp. 1142–1145, 2015.
- [7] M. J. Vansteensel, E. G. Pels, M. G. Bleichner, M. P. Branco, T. Denison, Z. V. Freudenburg, P. Gosselaar, S. Leinders, T. H. Ottens, M. A. Van Den Boom, *et al.*, “Fully implanted brain–computer interface in a locked-in patient with als”, *New England Journal of Medicine*, vol. 375, no. 21, pp. 2060–2066, 2016.
- [8] C. Pandarinath, P. Nuyujukian, C. H. Blabe, B. L. Sorice, J. Saab, F. R. Willett, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, “High performance communication by people with paralysis using an intracortical brain-computer interface”, *Elife*, vol. 6, e18554, 2017.

- [9] J. S. Brumberg, K. M. Pitt, and J. D. Burnison, “A noninvasive brain-computer interface for real-time speech synthesis: The importance of multimodal feedback”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 874–881, 2018.
- [10] D. A. Moses, M. K. Leonard, J. G. Makin, and E. F. Chang, “Real-time decoding of question-and-answer speech dialogue using human cortical activity”, *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.
- [11] F. R. Willett, D. T. Avansino, L. R. Hochberg, J. M. Henderson, and K. V. Shenoy, “High-performance brain-to-text communication via handwriting”, *Nature*, vol. 593, no. 7858, pp. 249–254, 2021.
- [12] M. P. Branco, Z. V. Freudenburg, E. J. Aarnoutse, M. G. Bleichner, M. J. Vansteensel, and N. F. Ramsey, “Decoding hand gestures from primary somatosensory cortex using high-density ecog”, *NeuroImage*, vol. 147, pp. 130–142, 2017.
- [13] M. G. Bleichner, Z. V. Freudenburg, J. M. Jansma, E. J. Aarnoutse, M. J. Vansteensel, and N. F. Ramsey, “Give me a sign: Decoding four complex hand gestures based on high-density ecog”, *Brain Structure and Function*, vol. 221, pp. 203–216, 2016.
- [14] E. West, *Which is more efficient: Talking or signing?*, <https://www.britishdeafnews.co.uk/which-is-more-efficient-talking-or-signing/>, Accessed: 2022-11-05.
- [15] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, “Direct cortical control of 3d neuroprosthetic devices”, *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.
- [16] S. V. Hiremath, W. Chen, W. Wang, S. Foldes, Y. Yang, E. C. Tyler-Kabara, J. L. Collinger, and M. L. Boninger, “Brain computer interface learning for systems based on electrocorticography and intracortical microelectrode arrays”, *Frontiers in integrative neuroscience*, vol. 9, p. 40, 2015.
- [17] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, “High-performance neuroprosthetic control by an individual with tetraplegia”, *The Lancet*, vol. 381, no. 9866, pp. 557–564, 2013.
- [18] P. M. Rossini, M. Noris Ferilli, and F. Ferreri, “Cortical plasticity and brain computer interface”, *Eur J Phys Rehabil Med*, vol. 48, no. 2, pp. 307–312, 2012.
- [19] C. M. Stinear, W. D. Byblow, M. Steyvers, O. Levin, and S. P. Swinnen, “Kinesthetic, but not visual, motor imagery modulates corticomotor excitability”, *Experimental brain research*, vol. 168, no. 1, pp. 157–164, 2006.

- [20] A. Heilinger, R. Ortner, V. La Bella, Z. R. Lugo, C. Chatelle, S. Laureys, R. Spataro, and C. Guger, “Performance differences using a vibro-tactile p300 bci in lis-patients diagnosed with stroke and als”, *Frontiers in Neuroscience*, vol. 12, p. 514, 2018.
- [21] C. Jeunet, C. Vi, D. Spelmezan, B. N’Kaoua, F. Lotte, and S. Subramanian, “Continuous tactile feedback for motor-imagery based brain-computer interaction in a multitasking context”, in *IFIP Conference on Human-Computer Interaction*, Springer, 2015, pp. 488–505.

Appendix A

Source Space Reduction for eLORETA

A.1 Simulated EEG Dataset

Figure A.1 shows the processing pipeline for the Simulated EEG Dataset. At the end of it, as explained in Section 3.1, we obtain a matrix *data* of 1000×108 , with 1000 the length of the time series and 108 the number of channels in the New York Head Model.

A.1.1 Initialisation

After implementing the 3 brain sources, we use cross-spectrum filtering on the generated data. This is a required parameter for BC-VARETA. We used *data2cs_event*, a Matlab function called *data2cs_event.m* that was developed by Guido Nolte as part of the MEG & EEG Toolbox of Hamburg University [1], available from the Simulation framework of Anzolin et al.: https://github.com/paolop21/simulation_source_connectivity. A high-level description is given in Algorithm 2; *fs* is the sample frequency, which is set to 100; *mean* and *real* are default Matlab functions. The resulting *Svv* is sized 108×108 .

A.1.2 Methods

LCMV

The *mkfilt_lcmv* function we used is the one developed by Guido Nolte as part of the MEG & EEG Toolbox of Hamburg University [1], also available

Algorithm 2: Cross spectrum filtering

length of each segment to estimate the FFT
 $seglen \leftarrow fs;$
 length of one epoch (can be ignored here: only important for event-related analysis)
 $epleng \leftarrow 2 * fs;$
 shift of each segment. $seglen/2$ leads to 50% overlap of segments
 $segshift \leftarrow seglen/2;$
 maximum frequency bin to estimate.
 $maxfreqbin \leftarrow (seglen/2) + 1;$
 $para \leftarrow [];$
 average over segments
 $para.segave \leftarrow 1;$
 subtract average
 $para.subave \leftarrow 0;$
 $[cs, tmp, nave] \leftarrow$
 $\text{data2cs_event}(data, seglen, segshift, epleng, maxfreqbin, para);$
 $Svv \leftarrow \text{mean}(\text{real}(cs), 3);$

from the Simulation framework of Anzolin et al.: https://github.com/paolop21/simulation_source_connectivity. A high-level description is given in Algorithm 3; Svv is sized 108×108 and the $head_model$ 108×2004 ; abs , $diag$ and max are default Matlab functions.

Algorithm 3: LCMV

$[A, P, po] \leftarrow \text{mkfilt_lcmv}(head_model, Svv);$ $connect_iv \leftarrow P' * Svv * P;$
 $sources_iv \leftarrow abs(diag(connect_iv));$
 $sources_iv \leftarrow sources_iv/max(sources_iv(:));$
 $sources_iv(sources_iv < 0.01) = 0;$

eLORETA

The $mkfilt_eloreta2$ function was developed by Guido Nolte as part of the MEG & EEG Toolbox of Hamburg University [1] and also available from the Simulation framework from Anzolin et al.: https://github.com/paolop21/simulation_source_connectivity. A high-level description is given in Algorithm 4; abs , $diag$ and max are default Matlab functions.

We used the default value of 0.01 for γ . The $head_model$ is a 108×2004 sized matrix.

Algorithm 4: eLORETA

```
 $P \leftarrow \text{mkfilt\_eloreta2}(\text{head\_model}, \gamma); \text{connect\_iv} \leftarrow P' * Svv * P;$ 
 $\text{sources\_iv} \leftarrow \text{abs}(\text{diag}(\text{connect\_iv}));$ 
 $\text{sources\_iv} \leftarrow \text{sources\_iv}/\max(\text{sources\_iv}(:));$ 
 $\text{sources\_iv}(\text{sources\_iv} < 0.01) = 0;$ 
```

BC-VARETA

The source code for Structured Sparse Bayesian Learning and BC-VARETA can be found on Github: <https://github.com/dpazlinares/BC-VARETA-simpack>. A high-level description is given in Algorithm 5.

Parameter $Nseg$ is initialised to 80, the default value; Svv is sized 108×108 and the head_model 108×2004 ; vertices (2004×3) and faces (4000×3) are the mesh information of the head model; abs , diag and \max are Matlab default functions.

Algorithm 5: BC-VARETA

```
[ $\Theta_{JJ}, SJJ, indms$ ]  $\leftarrow$ 
 $\text{bcvareta}(Svv, \text{head\_model}, Nseg, \text{vertices}, \text{faces});$ 
 $\text{sources\_iv} \leftarrow \text{zeros}(\text{length}(\text{head}_\text{model}), 1);$ 
 $\text{sources\_iv}(indms) \leftarrow \text{abs}(\text{diag}(SJJ));$ 
 $\text{sources\_iv} \leftarrow \text{sources\_iv}/\max(\text{sources\_iv}(:)); ind_zr \leftarrow \text{sources\_iv} < 0.01;$ 
 $\text{sources\_iv}(ind_zr) \leftarrow 0;$ 
```

Sparse eLORETA

The source code for Structured Sparse Bayesian Learning (*screening_ssbl* and *screening*) and BC-VARETA can be found on Github: <https://github.com/dpazlinares/BC-VARETA-simpack>. The *mkfilt_eloreta2* function is developed by Guido Nolte as part of the MEG & EEG Toolbox of Hamburg University [1] and also available from the Simulation framework of Anzolin et al.: https://github.com/paolop21/simulation_source_connectivity.

A high-level description is given in Algorithm 6. Parameter $Nseg$ is initialised to 80, the default value; parameter κ is the sparsity constraint described in Section 3; Svv is sized 108×108 and the head_model 108×2004 ; vertices is sized 2004×3 and faces 4000×3 , the mesh parameters of the head model.

Algorithm 6: Sparse eLORETA

```

 $Nseg \leftarrow 80;$ 
 $[p, qfull] \leftarrow size(head\_model);$ 
 $groups \leftarrow [];$ 
for  $ii \leftarrow 1$  to  $qfull$  by 1 do
|  $groups\{ii\} \leftarrow ii$ 
end for
 $indana \leftarrow 1 : qfull;$ 
 $miu \leftarrow screening\_ssbl(Svv, head\_model, Nseg, groups);$ 
 $[indms] \leftarrow screening(miu, kappa * qfull, vertices, faces, indana);$ 
 $head\_model\_red \leftarrow head\_model(:, indms);$ 
 $P \leftarrow mkfilt\_eloreta2(head\_model\_red, \gamma);$ 

```

A.1.3 Medoid Clustering

The Matlab function *kmedoids* was used to generate the 3 cluster result of Section 3.2. The clustering algorithm is described in Algorithm 7. *sources_iv* is the output from the various source localization methods with size $\#Dipoles \times 1$, in our case 2004×1 ; *epsilon*, an activity threshold, was set at 0.1.

locs_hm is a matrix sized 2004×3 containing the *x*, *y* and *z* coordinates of all dipoles of the head model; *all_ind* is a matrix of 2004×1 containing a list of all indices of all dipoles of the head model; *sort* is a Matlab default function.

Algorithm 7: Clustering Algorithm

```

 $ind\_act \leftarrow all\_ind(sources\_iv > epsilon);$ 
 $locs\_act \leftarrow locs\_hm(ind\_act, :);$ 
 $X \leftarrow locs\_act;$ 
 $[idx, C, sumd] \leftarrow kmedoids(locs\_act, 3);$ 
 $[out, sor] \leftarrow sort(sumd);$ 
 $cls\_1 \leftarrow X(idx == sor(1), :);$ 
 $cls\_2 \leftarrow X(idx == sor(2), :);$ 
 $cls\_3 \leftarrow X(idx == sor(3), :);$ 

```

A.2 Imaginary Movement EEG Dataset

A.2.1 Head Model

The Cuban Project head model we used can be retrieved from Github: <https://github.com/egmoreira/BC-VARETA-toolbox>. Loading the head model is described in Algorithm 8.

Algorithm 8: Head Model Loading

```
load(fullfile(currdir,'CUBAN_DATA_6k_15k_INFO'));
load(fullfile(currdir,'EEG_data'));
load(fullfile(currdir,'test.mat'));
locs_hm = S_6k.Vertices; all_ind = 1 : size(S_6k.Vertices,1);
head_model = K_6k; vertices = S_6k.Vertices; faces = S_6k.Faces;
names_hm = elect_58_343.label';
```

A.2.2 Methods

LCMV

A high-level description is given in Algorithm 3; Svv is sized 53×53 and the $head_model$ 53×6003 ; abs , $diag$ and max are default Matlab functions.

eLORETA

A high-level description is given in Algorithm 4; $head_model$ is a 53×6003 matrix and abs , $diag$ and max are default Matlab functions; for γ we used its default value of 0.01 .

BC-VARETA

A high-level description is given in Algorithm 5. Parameter $Nseg$ is initialized to 80, its default value; parameter Svv is sized 53×53 and $head_model$ 53×6003 ; $vertices$ is sized 6003×3 and $faces$ 11984×3 , the mesh parameters of the head model; abs , $diag$ and max are default Matlab functions.

Sparse eLORETA

A high-level description is given in Algorithm 6; $Nseg$ is initialized to 80, its default value; parameter κ is discussed in Section 3; Svv is sized 53×53 , and $head_model$ 53×6003 ; $vertices$ (6003×3) and $faces$ (11984×3) are the mesh parameters of the head model.

A.3 Cuban EEG Dataset

A.3.1 Methods

LCMV

A high-level description is given in Algorithm 3; parameter Svv is sized 58×58 and the $head_model$ 58×6003 ; abs , $diag$ and max are default Matlab functions.

eLORETA

A high-level description is given in Algorithm 4; *abs*, *diag* and *max* are default Matlab functions; for parameter γ we used its default value of 0.01; *head_model* is a 58×6003 matrix.

BC-VARETA

A high-level description is given in Algorithm 5. Parameter *Nseg* is initialized to 80, its default value; parameter *Svv* is sized 58×58 , the *head_model* 58×6003 ; *vertices* is sized 6003×3 and *faces* 11984×3 , the mesh parameters of the head model; *abs*, *diag* and *max* are default Matlab functions.

Sparse eLORETA

A high-level description is given in Algorithm 6. Parameter *Nseg* is initialized to 80, its default value; parameter κ is discussed in Section 3; *Svv* is sized 58×58 and the *head_model* 58×6003 . *vertices* is sized 6003×3 and *faces* 11984×3 , the mesh parameters of the head model.

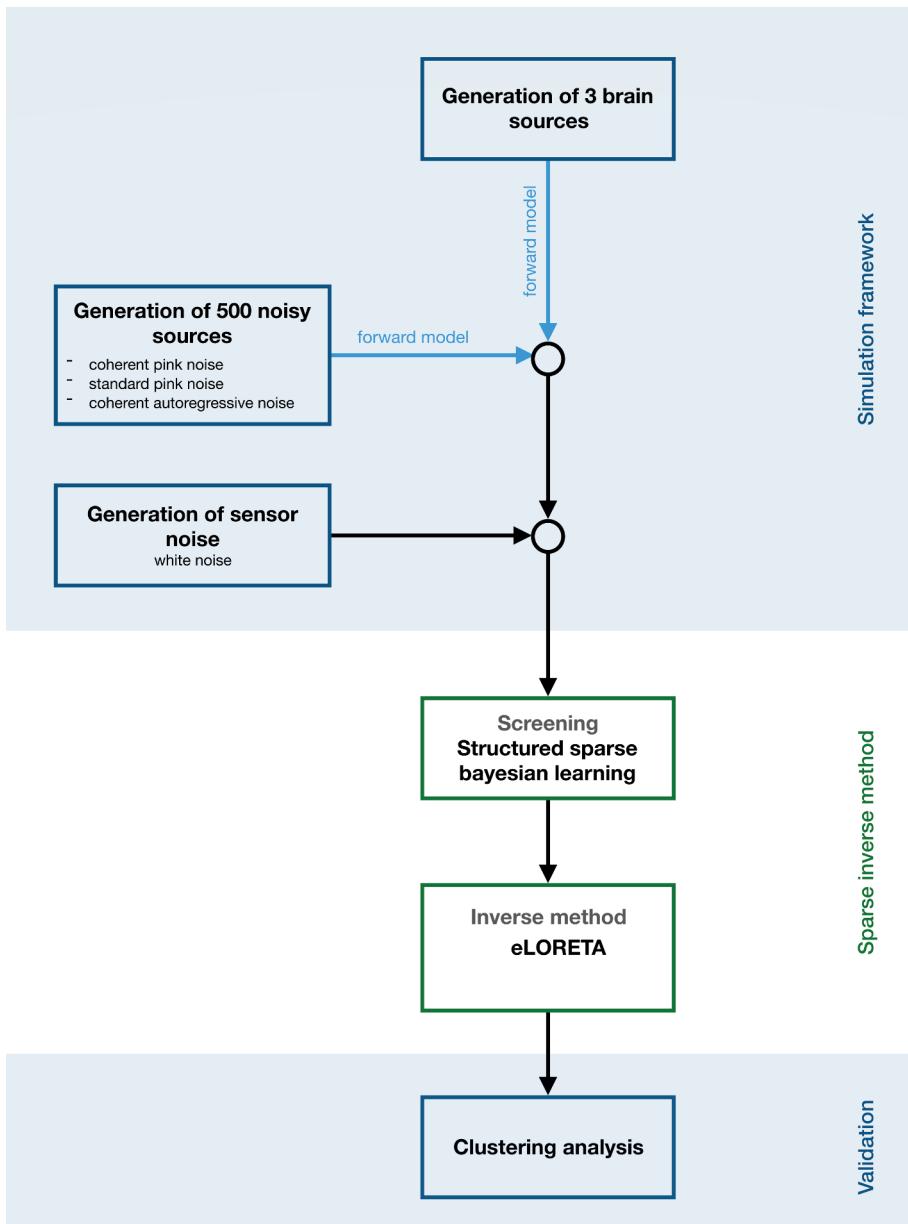


Figure A.1: Processing pipeline. Schematic rendition in the case of the simulated EEG dataset.

Bibliography

- [1] F. Freyer, M. Reinacher, G. Nolte, H. R. Dinse, and P. Ritter, “Repetitive tactile stimulation changes resting-state functional connectivity—implications for treatment of sensorimotor decline”, *Frontiers in human neuroscience*, vol. 6, p. 144, 2012.

Appendix B

Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data

B.1 Supplementary Material

Predictor	Sum of Squares	df	Mean Square	H	p-Value
Dipole condition	577.60	1		4.53	0.033
Conv2D model	2045.35	3		16.04	0.001
Interaction	350.15	3		2.75	0.432
Within	1999.40	32			
Total	4972.50	39	127.5		

Table B.1: Scheirer–Ray–Hare Test using Sensitivity as a criterion. Alpha level = 0.05. Significant results shown in bold. Models: Conv2D, TS = 2, kernel size 4×2 versus 4×4 , Conv2D, TS = 3, kernel size 4×3 versus 4×4 , with TS denoting the amount of time series. Model ($H(3,32) = 16.04, p = 0.001$) and dipole condition ($H(1,32) = 4.53, p = 0.03$) have significant effects on Sensitivity. Post hoc Mann–Whitney U tests (Bonferroni-corrected alpha level = 0.008) revealed that Sensitivity was significantly higher for Conv2D, TS = 2 (Mdn = 0.67, 0.67), versus Conv2D, TS = 3 (Mdn = 0.00, 0.17), $p = 0.006, p = 0.007$ for differing kernel sizes.

Appendix C

Finger movement and coactivation predicted from intracranial brain activity using extended block-term tensor regression

C.1 Supplementary Material

	Thumb	Index	Middle	Ring	Pinky
Thumb	/	0.72 / 0.05 (0.17)	0.73 / 0.04 (0.13)	0.81 / 0.04 (0.18)	0.7 / 0.05 (0.14)
Index	0.64 / 0.06 (0.2)	/	0.78 / 0.36 (0.16)	0.73 / 0.14 (0.34)	0.71 / 0.01 (0.13)
Middle	0.74 / 0.16 (0.25)	0.74 / 0.62 (0.24)	/	0.62 / 0.25 (0.22)	0.62 / 0.24 (0.34)
Ring	0.74 / 0.06 (0.29)	0.67 / 0.26 (0.23)	0.72 / 0.19 (0.26)	/	0.78 / 0.42 (0.67)
Pinky	0.74 / 0.06 (0.23)	0.43 / 0.05 (0.22)	0.54 / 0.09 (0.2)	0.82 / 0.18 (0.31)	/

Table C.1: Row-wise: Subject 1's unintended finger movement accuracies for for eBTTR / BTTR; between brackets the ratio between the unintended and intended (cued) finger flexions; the cued finger is marked by “-“, the accuracy listed in Table 2 in the results section. The finger flexion ratios show to what extent the non-cued finger moves along with the cued finger. Note the difference between thumb and index finger and the significant difference in accuracy between eBTTR and BTTR's predicted unintended movements. Accuracies are expressed by the Pearson correlation coefficient between predicted and actual finger movement trajectories.

	Thumb	Index	Middle	Ring	Pinky
Thumb	/	0.71 / 0.13 (0.27)	0.72 / 0.05 (0.15)	0.74 / 0.27 (0.24)	0.72 / 0.25 (0.4)
Index	0.73 / 0.04 (0.28)	/	0.56 / 0.33 (0.2)	0.73 / 0.17 (0.24)	0.6 / 0.01 (0.26)
Middle	0.34 / 0.05 (0.24)	0.36 / 0.2 (0.34)	/	0.35 / 0.22 (0.32)	0.38 / 0.15 (0.22)
Ring	0.62 / 0.02 (0.35)	0.64 / 0.23 (0.36)	0.67 / 0.17 (0.48)	/	0.75 / 0.26 (0.64)
Pinky	0.73 / 0.07 (0.23)	0.61 / 0.06 (0.19)	0.64 / 0.05 (0.18)	0.64 / 0.19 (0.35)	/

Table C.2: . Idem to Table C.1 but for Subject 2.

	Thumb	Index	Middle	Ring	Pinky
Thumb	/	0.62 / 0.15 (0.3)	0.73 / 0.24 (0.27)	0.71 / 0.24 (0.32)	0.7 / 0.25 (0.29)
Index	0.73 / 0.14 (0.4)	/	0.76 / 0.43 (0.62)	0.83 / 0.27 (0.5)	0.7 / 0.31 (0.46)
Middle	0.64 / 0.16 (0.25)	0.64 / 0.32 (0.51)	/	0.62 / 0.36 (0.58)	0.72 / 0.24 (0.4)
Ring	0.62 / 0.12 (0.22)	0.64 / 0.33 (0.36)	0.72 / 0.37 (0.58)	/	0.75 / 0.43 (0.86)
Pinky	0.63 / 0.17 (0.17)	0.51 / 0.16 (0.21)	0.74 / 0.25 (0.33)	0.64 / 0.19 (0.54)	/

Table C.3: Idem to Table C.1 but for Subject 3.

Curriculum Vitae

Axel Faes

Email axel.faes@gmail.com

Github TheAxeC

Google Scholar Axel Faes

KU Leuven Who's Who 00117812

Educational background

Doctoral Programme in Biomedical Science (PhD) in Computational Neuroscience

KU Leuven Leuven, Belgium
Sep. 2018 - Sep. 2022
Cognitive and Molecular Neuroscience

Advanced Master of Science in Engineering (M.Sc.) in Artificial Intelligence (76% - Cum Laude)

KU Leuven Leuven, Belgium
Sep. 2017 - Jul. 2018
Engineering and Computer Science

Honoursprogramme of the Faculty of Engineering Science

KU Leuven Leuven, Belgium
Sep. 2016 - Oct. 2018
Research track - 18 ECTS (2 projects of 9 ECTS) completed in 2016 - 2017.

Master of Science in Engineering (M.Sc.) in Computer Science (Burgelijk Ingenieur - ir.) (76% - Cum Laude)

KU Leuven Leuven, Belgium
 Sep. 2016 - Sep. 2018
 Artificial Intelligence & Theoretical Computer Science

Bachelor of Science (B.Sc.) in Computer Science (79% - Magna Cum Laude)

UHasselt Hasselt, Belgium
 Sep. 2013 - Jul. 2016
 Physics and General courses

Business Summer School: United in Manchester (0739)

The University of Manchester Manchester, UK
 Jul. 2015 - Aug. 2015
 International Business

Honors & Awards

Mar. 2018	Finalist, Cyber Security Challenge	Brussels, Belgium
Sep. 2017	3rd place, ICFP 2017 Student Research Competition	Oxford, UK
Jul. 2016	Bachelor Award, in Computer Science	UHasselt, Belgium
May. 2016	3rd place, ACM CHI 2016 Student Design Competition (Interaction Design and User Experience.)	San Jose, CA, USA
Feb. 2016	2nd place, BeGDC (Belgian Game Development Championship)	Brussel, Belgium
Jan. 2016	IELTS, Academic Module (8.0/9.0)	Brussel, Belgium

Work & Research Experience

Phd Candidate (FWO-Aspirant Fellowship)

KU Leuven Leuven, Belgium
 Oct. 2018 - Current

- Project: "Graph-based model of information diffusion in the human brain for studying event-related potentials"
- Promoter: Prof. Marc van Hulle
- Group: Computational Neuroscience, Laboratory for Neuro-and Psychophysiology, KU Leuven

Student Job: Creating System Identification course

KU Leuven

Leuven, Belgium

Sep. 2017 - Current

- Faculty of Engineering Science: ESAT (Electrical Engineering)
- Research group: STADIUS

Research Assistant: design of type-&-effect system for Eff based on row polymorphism

KU Leuven

Leuven, Belgium

Apr. 2017 - Oct. 2017

- Faculty of Engineering Science: Computer Science
- Research group: DTAI
- Part of the Honoursprogramme of the Faculty of Engineering Science (research track).
- Topic: Development of an row-based type-&-effect system for the Eff programming language

Research Assistent: efficient compilation of algebraic effect handlers

KU Leuven

Leuven, Belgium

Sep. 2016 - Apr. 2017

- Faculty of Engineering Science: Computer Science
- Research group: DTAI
- Part of the Honoursprogramme of the Faculty of Engineering Science (research track). My project is part of the C1 project: Algebraic Effect Handlers: Harnessing the Fundamental Power of Effects. Eff is a functional programming language that uses handlers to handle all kinds of effects. These effects could be I/O, exceptions, user-defined, etc. My task is to design, implement, benchmark and formally proof new optimisations in the Eff compiler. The compiler is written in OCaml.

Web Performance Research Internship

Expertise centre for Digital Media (EDM),
UHasselt

Hasselt, Belgium

Jul. 2016 - Sep. 2016

- I worked on the iMinds PRO-FLOW project.
- My work involved creating multiple usecases to measure website performance. The main focus is on the difference between the http versions (http1.1, https, http2).
- During the project, I had to utilise multiple servers, maintain and extend the nodejs framework used to measure website performance, and manually optimize commercial websites using PHP, JS, HTML and CSS.

Summer Research Internship Physical Computing

Expertise centre for Digital Media (EDM),
UHasselt

Hasselt, Belgium

Aug. 2015 - Sep. 2015

- Work on a project which focuses on the interaction between a human entity and a drone, aswell as interaction between the drone and virtual objects. This project is written in C++, used the Optitrack motion capture and a custom created drone.

Extracurricular Activities

KU Leuven

DjangoGirls coach

Leuven, Belgium

Mar. 2018 - Current

- We inspire women to fall in love with programming.
- Django Girls organize free Python and Django workshops, create open sourced online tutorials and curate amazing first experiences with technology.

KU Leuven

Student RepresentativeLeuven, Belgium
Sep. 2017 - Current

- POC of Master Computer Science Engineering
- POC of Advanced Master Artificial Intelligence
- Member of Departmental council of Computer Science
- Member of Department board of Computer Science
- Member of Faculty council of Engineering Science

CoderDojo Belgium

Coach

Uassel, PXL

Sep. 2014 - Current

- Teach children how to program utilising Scratch, Python, Minecraft and Lego mindstorm.

Student Council

Member

UHasselt

Aug. 2015 - Aug. 2016

- Member of Board of Education
- Member of Faculty Council
- Member of Board of Student Facilities
- Member of Diversity Commission
- Temporary representative in VVS (Vlaamse Vereniging van Studenten vzw)

Student Council

Member

UHasselt

Mar. 2015 - Aug. 2015

- Member of Board of Education
- Member of Faculty Council
- Member of Board of Student Facilities

UHAsselt

Student Representative

Hasselt, Belgium
Sep. 2013 - Jul. 2016

- Representing students interests in a Computer Science education context.
- This meant discussing with the university in order to improve teaching, learning, assessment and academic services.
- Representing Computer Science education for high school students

Natural Languages

English Fluent (IELTS: 8.0/9.0)

Dutch Mothertongue

French Basic Knowledge

Projects

An Information Theoretical Approach to EEG Source-Reconstructed Connectivity

Advanced Master's Thesis

Leuven, Belgium
Feb. 2018 - Jul. 2018

- This thesis takes an information theoretical approach, which concerns model-free, probability based methods such as Conditional Mutual Information, Directed Information, and Directed feature information.
- 17/20

Algebraic Subtyping for Algebraic Effects and Handlers

Master's Thesis

Leuven, Belgium
Feb. 2018 - Jul. 2018

- Extending Algebraic Subtyping to incorporate support for algebraic effects and handlers. Final score - 19/20

Reinforcement Learning Agent in Google Deepmind's StarCraft II Framework - CSAI

Developer

Leuven, Belgium
Feb. 2018 - Jul. 2018

- Implement several learning algorithms in PySC2

Software Architecture course - Project

Developer

Leuven, Belgium
Feb. 2017 - Jul. 2017

- Project made for the course 'Software Architecture'. The goal was to design a software architecture in UML for a IoT-platform concerning pluggable sensors. The platform allows storage of customer data and the use of third party applications for data analytics. Final score - 18/20

ICAL parser for KU Leuven schedules

Lead Developer

Leuven, Belgium
Aug. 2016 - Current

- An nodejs application to create an iCalender file for courses at KU Leuven. Allows the creation of a schedule containing courses from different masters and the option to ignore events.

Machine learning techniques for flow-based network intrusion detection systems

Bachelor's Thesis

Hasselt, Belgium
Feb. 2016 - Jul. 2016

- The thesis gives an overview of how machine learning algorithms could be used for intrusion detection using only IP Flows. The system has been used to detect intrusions in Cegeka Hasselt Datacenter network.

Software engineering: Search and Recommendation System

Team Member

Hasselt, Belgium
Feb. 2016 - Jul. 2016

- A search and recommendation system for VoD (Video on Demand) for Androme. The system is currently being used in production in the Nebula project. Both Content-Based Recommendations and Collaborative filtering techniques were implemented. Made in a team of 5 (Pieter Teunen, Luuk Raaijmakers, Brent Berghmans, Axel Faes, Matthijs Kaminski, Wouter Bollaert) utilising Java and the Spring framework. Final score - 15/20

TTUI: Household Survival

Researcher

Hasselt, Belgium
Sep. 2015 - Dec. 2015

- Project made for the class 'Technologies and Tools for User Interfaces'.
- A tower-defense style game written in Unity utilising Optitrack motion capture. The game combines the virtual world and reality, by allowing users to interact with the virtual world using real-world objects. Made by Brent Berghmans, Axel Faes and Matthijs Kaminski. Final score - 18/20

Cardinal: scripting language**Lead Developer**

Hasselt, Belgium
Jan. 2015 - Sep. 2015

- Cardinal is a small, fast, class-based, Object Oriented scripting language written in C. It is built upon the skeleton of an existing scripting language and shows how I can modify and improve existing software, as well as design new components to this software.
- New components include a debugger, an embedding API, multiple inheritance and a new module system.

United in Manchester**Team Leader**

Manchester, UK
Jul. 2015 - Aug. 2015

- A summer school which focuses on teamwork in cross-cultural and multidisciplinary teams, global product development and entrepreneurship. Our team developed a start-up idea on Food Management/Delivery system. Product pitch took place at the end of the course for feedbacks from professionals. Our team consisted of Axel Faes, Linh Chi Evelyn Phan, Reinaert Van de Cruys and Maria Barouh.

PSOPV: Visual Programming IDE**Developer**

Hasselt, Belgium
Feb. 2015 - Jul. 2015

- A Visual programming IDE created by Axel Faes & Matthijs Kaminski for a course of Hasselt University. The purpose of the IDE is to create 'black boxes' which can send events (signals packed with data) to each other. We take the idea of using drag-able blocks in a visual IDE and expand on it. Final score - 17/20

Publications

INTERNATIONAL JOURNAL PAPERS

- Axel Faes, Marc M. Van Hulle, “Finger movement and coactivation predicted from intracranial brain activity using extended Block-Term Tensor Regression”, Journal of Neural Engineering.
- [1] Axel Faes, Flavio Camarrone, Marc M. Van Hulle, “Single finger trajectory prediction from intracranial brain activity using Block-Term Tensor Regression with fast and automatic component extraction”, IEEE Transactions on Neural Networks and Learning Systems.
- [2] Axel Faes, Aurelie de Borman, Marc M. Van Hulle, “Source space reduction for eLORETA”, Journal of Neural Engineering.
- [3] Axel Faes, Iris Vantieghem, Marc M. Van Hulle, “Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data”, Applied Sciences.
- [4]

CONFERENCE PAPERS

- Robin Marx, Maarten Wijnants, Peter Quax, Axel Faes, Wim Lamotte, “Web Performance Characteristics of HTTP/2 and comparison to HTTP/1.1”, International Conference on Web Information Systems and Technologies, pg 87-114.
- [5]
- Robin Marx, Peter Quax, Axel Faes and Wim Lamotte, “Concatenation, embedding and sharding: Do HTTP/1 performance best practices make sense in HTTP/2?”, WEBIST 2017 - Proceedings of the 13th International Conference on Web Information Systems and Technologies.
- [6]

EXTENDED ABSTRACTS

- Axel Faes and Tom Schrijvers, “Towards a Core Language with Row-Based Effects for Optimised Compilation”, International Conference on Functional Programming 2017 Student Research Competition.
- [7] Kashyap Todi, Brent Berghmans, Axel Faes and Matthijs Kaminski, “Purpose-Centric Appropriation of Everyday Objects as Game Controllers”, CHI EA ’16: Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems. Late Breaking Work.
- Kashyap Todi, Donald Degraen, Brent Berghmans, Axel Faes, Matthijs Kaminski and Kris Luyten, “Household Survival: Immersive Room-Sized Gaming Using Everyday Objects as Weapons”, CHI EA ’16: Extended Abstracts of the SIGCHI Conference on Human Factors in Computing Systems. Student Game Competition.
- [9]

THESIS

- [10] Axel Faes, “An Information Theoretical Approach to EEG Source-Reconstructed Connectivity”, Advanced Master’s Thesis 2018.
- [11] Axel Faes, “Algebraic Subtyping for Algebraic Effects and Handlers”, Master’s Thesis 2018.
- [12] Axel Faes, “Machine learning techniques for flow-based network intrusion detection systems”, Bachelor’s thesis 2016.

POSTERS

- Axel Faes and Tom Schrijvers, “Towards a Core Language with Row-Based Effects for Optimised Compilation”, International Conference on Functional Programming 2017 Student Research Competition.
- [13]

OTHER PUBLICATION

- Matija Pretnar, Amr Hany Shehata Saleh, Axel Faes and Tom Schrijvers, “Efficient compilation of algebraic effects and handlers”, 2017 - CW Reports, CW708, 32 pp. Leuven, Belgium: Department of Computer Science, KU Leuven..
- [14]

TALKS, PRESENTATIONS AND OTHER MEDIA

“Honours student Axel Faes wins bronze
Sep. 19, 2017 medal in ACM SIGPLAN”, KU Leuven,
Department of Computer Science.

“Student Axel Faes wins bronze medal in
the ACM SIGPLAN Student Research
Sep. 19, 2017 Competition in ICFP conference”, KU
Leuven, Department of Computer Science,
DTAI.

Publications

Publications

INTERNATIONAL JOURNAL PAPERS

- Axel Faes, Marc M. Van Hulle, “Finger movement and coactivation predicted from intracranial brain activity using extended Block-Term Tensor Regression”, Journal of Neural Engineering.
 - [1] Axel Faes, Flavio Camarrone, Marc M. Van Hulle, “Single finger trajectory prediction from intracranial brain activity using Block-Term Tensor Regression with fast and automatic component extraction”, IEEE Transactions on Neural Networks and Learning Systems.
 - [2] Axel Faes, Aurelie de Borman, Marc M. Van Hulle, “Source space reduction for eLORETA”, Journal of Neural Engineering.
 - [3] Axel Faes, Iris Vantieghem, Marc M. Van Hulle, “Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data”, Applied Sciences.
 - [4]

Appendix D

Acknowledgements

The candidate was supported by the Belgian Fund for Scientific Research – Flanders (FWO 1157019N).

Appendix E

Personal Contribution

The work reported in this thesis was primarily done by the candidate, in some cases benefitting from collaborations as follows:

- A. De Borman contributed in the data analysis for Chapter 2: Source space reduction for eLORETA
- Iris Vantieghem contributed in the data analysis and development of Conv2D for Chapter 3: Neural Networks for Directed Connectivity Estimation in Source-Reconstructed EEG Data
- Flavio Camarrone contributed equally to the development of BTTR for Chapter 4: Single Finger Trajectory Prediction From Intracranial Brain Activity Using Block-Term Tensor Regression With Fast and Automatic Component Extraction
- All co-authors reviewed the publications prior to publication
- Wannes Meert developed the Latex template behind the PhD thesis

Appendix F

Conflict of Interest

All co-authors declare no conflict-of-interest for any of the abovementioned contributions.

FACULTY OF MEDICINE
DEPARTMENT OF NEUROSCIENCE
LABORATORY FOR NEURO- AND PSYCHOPHYSIOLOGY

Herestraat 49

3000 Leuven

axel.faes@kuleuven.be

www.website.kuleuven.be



