A NON-INVASIVE BRAIN COMPUTER INTERFACE DECODER FOR GAIT

by

Sho Nakagome

A dissertation submitted to the Department of Electrical & Computer Engineering
Cullen College of Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Electrical & Computer Engineering

Chair of Committee: Jose L. Contreras-Vidal
Committee Member: Saurabh Prasad
Committee Member: David Mayerich
Committee Member: Hien Nguyen
Committee Member: Luca Pollonini

University of Houston
May 2020

Acknowledgements

First, I would like to thank my advisor, Dr. Jose L. Contreras-Vidal, for his guidance and support throughout my Ph.D. Without his guidance and support, I could not have accomplished this journey. Second, I would also like to thank all the members of the lab for their encouragement and friendship. Most of my learning and insights came from the talking and discussion with our people in the laboratory. I would like to specifically acknowledge Dr. Phat Luu, Dr. Yongtian He, Dr. Justin A. Brantley, Dr. Fangshi Zhu, Andrew Paek, and Akshay S. Ravindran, for their direct contributions to work presented in this dissertation and associated discussion regarding the thesis. Third, I would also like to acknowledge Dr. Saurabh Prasad, Dr. David Mayerich, Dr. Hien Nguyen, and Dr. Luca Pollonini, for participating as committee members of my dissertation. Specifically, thank you to Dr.Luca Pollonini for supporting me during the difficult times in my Ph.D. Fourth, thanks to all my friends across the world for encouraging me to be a better person through their attitude towards pursuing their own goals. At last, thank you to my parents, sister, and family for continuously believing in me.

This research was supported by award R01NS075889 from the National Institute of Neurological Disorders And Stroke (NINDS), National Science Foundation (NSF) award HCC 1302339, NSF IUCRC BRAIN award 1650536, and Mission Connect - A TIRR Foundation. We are also grateful for the support of the Core facility for Advanced Computing and DataScience at the University of Houston (CACDS) for assistance and providing computation resources required in this work.

Abstract

Brain Computer Interface (BCI) systems enable control of machines and computers using signals extracted from the brain, such as data recorded using electroencephalography (EEG). Naturally, this technology is expected to help people with disabilities, such as lost speech or motor impairment, by providing an alternative approach to interact with the world. Being able to walk is one of the most fundamental human functions, and BCIs could help those with walking impairment by providing direct control of an exoskeleton directly from brain signals. The most crucial part of building such a system is the neural decoding–i.e., the specific algorithm that translates neural signals into movement signals. Developing an effective neural decoding model does not only provide accurate control of the device, but could also open a new path towards understanding the neural representation of gait. A wide variety of algorithms have been proposed for neural decodings, such as linear regression, kalman filters, and artificial neural networks. However, there is a lack of rigorous comparisons of different decoding models and parameter choices. Furthermore, it is unclear how well each of these models will generalize to new data from either new environments or different subjects. This dissertation thesis aims to investigate those issues by: 1) Benchmarking the proposed models and understanding the representation of the brain during gait and 2) Study ways to generalize the model. In the first specific aim, we showed that neural networks not only performed better than conventional methods when trained within a specific walking environment, but resulted in models that were robust to external disturbances such as channel distortion. In the second aim, we showed intra- subject decoding works in all the combinations (e.g., inter-subject decoding of different terrains, level ground walking only, treadmill walking, etc.), but inter- subject decoding only works for electromyography (EMG) to kinematics decoding. To deal with this problem, several methods were used to improve inter- subject decoding. Of these methods, transfer learning achieved the most promising results. The work in this dissertation contributes to a greater understanding of the decoding

models and their performance/generalizability on non-invasive gait decoding.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Neural decoding in Brain Computer Interface (BCI) for gait

Neural decoding is a way to predict measurable outputs from neural signals for the purposes of solving engineering problems and understanding the brain [1]. There are two types of neural decoding: 1) Classification and 2) Regression. Classification based neural decoding tries to predict categorical outputs (e.g., different states, cues, etc.) and will not be the focus of this dissertation. Regression based neural decoding, otherwise known as continuous decoding, tries to predict continuous outputs (e.g., predicting kinematics, hand trajectory, etc.). A conceptual illustration of continuous decoding is presented in Figure 1.1. Continuous neural decoding started as a way to map neural signals to kinematics, effectively treating this as a linear regression problem [2]. One of the earliest studies showed a simple linear filter (Wiener filter) could be used to decode reach and grasp kinematics in monkeys from invasively recorded neural signals [3]. Another approach was to use a Kalman filter (KF) for neural decoding [4]. In fact, previous studies achieved fairly accurate decoding performance using a KF with updating rules for a continuous target-to-target task [5, 6]. Many improvements were made in the past decade, where various algorithms and experimental designs were pursued in both invasive and non-invasive BCIs (See recent reviews [7, 8, 9]).

BCI is a novel approach to utilize neural decoding to control various outputs [11, 12, 13]. Recent advances in BCI have demonstrated the feasibility of neural decoding

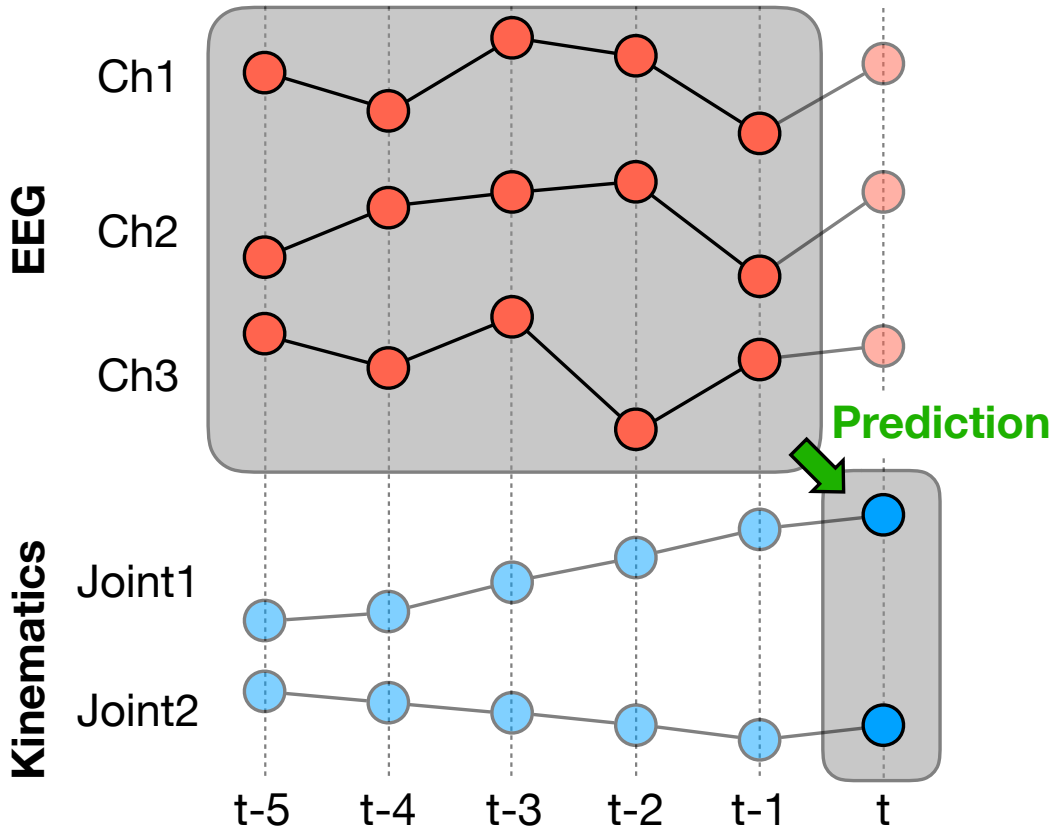

Figure 1.1: Example decoding schematic for tap size = 5. Illustration of the neural decoding schematic. Input channels will be used as inputs to predict the output channels one time sample ahead. Adapted from [10] CC BY 4.0).

using non-invasive technologies, such as electroencephalography (EEG), coupled with machine learning (ML) algorithms [14]. EEG enables the monitoring and utilization of neural signals during gait (= walking pattern) due to its portability and high temporal resolution [15, 16]. Previous studies used EEG for control of exoskeletons [17, 18], and to study the brain during treadmill [19, 20] and overground walking [21]. These studies revealed the feasibility of utilizing EEG for neuroimaging and decoding kinematics to study the modalities of the brain, and to use the brain signals as a control signal for a brain machine interface (BMI) during gait (General illustration of closed-loop BCI is illustrated in Figure 1.2).

EEG-based BCI neural decoders have the potential to augment plasticity and facilitate rehabilitation for patients with motor disabilities [13, 22]. The primary component of these applications is the neural decoder because decoding performance determines the usability of the system. It is also important to be able to inform clinicians, doctors, and researchers about the importance of specific features as to why such systems are working [9]. There are many features of the EEG signal that can be used when decoding human movement, such as time and frequency domain features, channel locations, and channel and source domain features [11, 14]. However, the complexity and the number of possible options of hand-selected features makes it difficult to identify and quantify the most important features for each decoding task. This is often the case for lower limb studies because it is more challenging to perform lower limb experiments compared to upper limb studies, resulting in difficulty collecting a large amount of high quality data to test for certain features of importance. Also, it is challenging to understand the representation and underlying neural mechanisms of the brain. Recent advances in artificial intelligence could provide insight into this challenge [23]. Studies suggest that building a neural decoder using deep learning and deciphering the layers within the trained neural decoder could help us understand the underlying mechanisms [9, 24]. Additionally, models, such as recurrent neural networks (RNN), could initiate hypothesis formulation [25], which

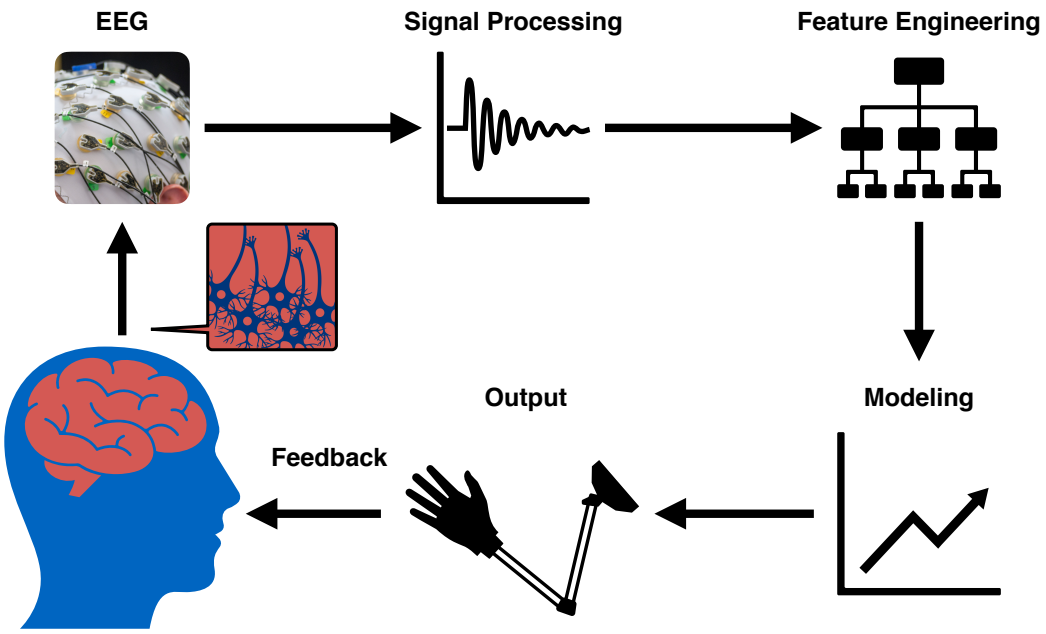

Figure 1.2: General illustration of closed-loop BCI

would enable us to study and quantify features that are relevant to the decoding task. Lastly, one of the most recent reviews also states the use of such modeling as a way to run a loop of modeling and experimenting in neuroscience [26].

Restoration of gait function has been a long-standing focus of rehabilitation research, and it is still an active research area to be explored [27, 28]. Although there are various approaches to this major issue, one of the promising approaches is to use a neural decoder to build applications and to understand the underlying mechanisms of the brain while walking [18, 29, 30]. However, the number of studies using decoders to understand the cortical networks during gait is still very limited due to the difficulty in performing experiments. Therefore, it is important to build a framework that could be open-sourced and be easily deployed into such applications that could be beneficial for studying patients' brains.

Currently, there are two major challenges in EEG-based gait decoding BCI: 1) Benchmarking and 2) Generalization. Each challenge is introduced in the following

sections with its significance to the problem.

1.1.1 Benchmarking and Modeling

With many algorithms and experimental designs proposed in the BCI field, it is essential to benchmark the models [24, 31, 32]. The importance of benchmarking is to reveal how one method compares to the other using multiple metrics. This could often reveal how ubiquitous methods, such as generalized linear models (GLM), may not be suitable in some cases [32]. For example, Glaser et al. illustrated the process of benchmarking a basic linear model by comparing the performance against more sophisticated models (otherwise known as benchmarking with baseline models) [24, 31].

1.1.2 Generalization and Transfer learning

Another main issue when conducting lower limb studies is the duration of the experiments [33]. Not only does the walking paradigm require more effort from the subjects, but also the preparation and experimental setup requires a significant amount of time. Additionally, current neural decoders require training every time before each experiment for good decoding performance, which is yet another reason that the experiments could take a longer time. In fact, a previous study investigated in this issue attempted to reduce recalibration with offline analysis [33]. Building a generalized neural decoder could significantly reduce the amount of time by providing a model that already has the required features and could be utilized on demand or with a small amount of time for fine-tuning. This could significantly reduce the burden of performing such experiments and could help the research community perform lower limb walking experiments more feasibly.

There are mainly two experimental paradigms when conducting gait related studies in EEG. The first is treadmill-based gait studies and the second is overground approaches. Although both experimental paradigms are walking related tasks, evidence shows that they are different, both dynamically and mechanically [34, 35, 36]. Moreover, it is often more challenging to perform overground walking experiments due to the necessity of larger space for the subject to move around and the difficulty in consistently acquiring high quality data [37]. This induces differences between the two paradigms that could lead to difficulty in interpreting the results that may only be true for one paradigm. Building a generalized neural decoder from treadmill walking that could then be applied to overground walking could potentially provide additional insights into this problem by identifying the commonality and differences in representations between the treadmill and overground walking.

Transfer learning and generalizing the decoder across trials, subjects, different modalities have yet to be explored. This is important not only because it could significantly reduce the amount of training time, but it could also provide another method to analyze the commonality and difference between two similar but slightly different paradigms (treadmill and overground). One of the latest studies shows that deep transfer learning using visually evoked potential based EEG is feasible in this aspect as the authors provided insights of better generalization across different trials, subjects, and paradigms [38].

1.2 Specific Aims

1.2.1 Aim 1: Benchmarking and Modeling

There are currently two trends in the neural decoding field. One main stream is to apply various signal processing techniques for artifact removal coupled with

simple linear or non-linear decoders. Another is to simply use an end-to-end deep learning technique with the aim of handling both artifact removal and decoding with high accuracy at the same time. A hybrid method combining both advantages could potentially lead to a preprocessing pipeline and a deep neural decoder that is robust to artifacts and can provide meaningful insights on the important features that are both easily interpretable and effective.

1.2.2 Aim 2: Generalization

One of the major problems with neural decoding is variability among trials and subjects. This often leads to the strategy of training a decoder for each subject, specifically tuning parameters for performance even before each trial for the same subject. The strength of deep learning is in its generalization. This property of deep learning, if applied successfully, could potentially solve this problem by generalizing the neural decoder across trials and subjects. We also seek the feasibility of transfer learning as a way to solve for the generalization problem.

Overall, these specific aims work in conjunction to solve an engineering problem with the goal of building a better data processing and neural decoding pipeline that has the potential to be a key towards a generalized lower limb neural decoder.

Chapter 2

An empirical comparison of neural networks and machine learning algorithms for EEG gait decoding

2.1 Abstract

Previous studies of Brain Computer Interfaces (BCI) based on scalp electroencephalography (EEG) have demonstrated the feasibility of decoding kinematics for lower limb movements during walking. In this computational study, we investigated offline decoding analysis with different models and conditions to assess how they influence the performance and stability of the decoder. Specifically, we conducted three computational decoding experiments that investigated decoding accuracy: 1) based on delta band time-domain features, 2) when downsampling data, 3) of different frequency band features. In each experiment, eight different decoder algorithms were compared including the current state-of-the-art. Different tap sizes (sample window sizes) were also evaluated for a real-time applicability assessment. A feature of importance analysis was conducted to ascertain which features were most relevant for decoding; moreover, the stability to perturbations was assessed to quantify the robustness of the methods. Results indicated that generally the Gated Recurrent Unit (GRU) and Quasi Recurrent Neural Network (QRNN) outperformed other methods in terms of decoding accuracy and stability. Previous state-of-the-art Unscented Kalman Filter (UKF) still outperformed other decoders when using smaller tap sizes,

with fast convergence in performance, but occurred at a cost to noise vulnerability. Downsampling and the inclusion of other frequency band features yielded overall improvement in performance. The results suggest that neural network-based decoders with downsampling or a wide range of frequency band features could not only improve decoder performance but also robustness with applications for stable use of BCIs.

2.2 Introduction

Brain Computer Interfaces (BCI) record, infer, and translate different parameters associated with movement from different types of brain signals to provide volitional control to prosthetic limbs, exoskeletons, computers, and even digital avatars. The part of the BCI which deciphers the user's motor intent from recorded brain activity is typically referred to as a neural decoder. Building high-performance neural decoders is important in four different aspects: 1) usability, 2) salient feature identification and quantification, 3) understanding of the underlying neural representations [24], and 4) a potential metric of neural function. First, BCI neural decoders based on scalp electroencephalography (EEG) are being designed for assistive and therapeutical applications for patients with motor disabilities in order to promote plasticity and facilitate rehabilitation [13, 22]. Thus, higher accuracy in decoding performance determines the usability of the system[31]. Second, many neural features (e.g., time and frequency domain features, channel locations, channel, and source domain features, to name a few [8, 14]) are likely to contain varying information about motor intent, and thus are candidates for decoding human movement. However, it is often difficult to identify and quantify important features given the complexities of performing lower limb experiments in people with gait disabilities limiting the amount of high-quality data. Third, decoder calibration is often focused on maximizing decoding accuracy while neglecting the explanatory power of the decoder itself. Thus, it has been difficult to advance understanding of the representation and underlying

neural mechanisms of the brain and recent advances in artificial intelligence could cast insight into this respect [23]. Studies suggest that building a neural decoder using deep learning could cast insights into this aspect by deciphering its neuron layers [24, 31]. This approach may enable us to study and quantify features that are relevant to the decoding task that could also help us understand the underlying mechanisms of the brain. At last, the accuracy of neural decoding could also reveal the amount of information explained by the model. It is a well-known fact that not only the cerebral cortex but also the cerebellum and spinal cord all play a crucial role in ambulatory movements. It is of interest to understand how much information we could extract from the cerebral cortex using non-invasive technology.

Restoration of gait function has been a long-standing focus of rehabilitation research and it is still an active research area to be explored [27, 28]. Although there are various approaches in this domain, one of the promising approaches is to use a neural decoder to build BCI applications and understand the underlying mechanisms of the brain associated with walking [18, 29, 30]. However, the number of studies using a decoder to understand the cortical networks during gait is still very limited. Therefore, it is important to build a decoding framework that could be open-sourced and be easily deployed into such applications that could be beneficial for studying the brain of patients with disabilities.

Another application of BCIs in this context is to incorporate the interface into real-time control of assistive devices that could help people with lower limb disabilities walk again. In this case, the accuracy of the decoding performance is crucial as it determines the usability of the system. The robustness of the algorithms is also important since the likelihood of decoding errors increases when the system is used in a real-world setting.

Neural signals are nonlinear and nonstationary [39]. However, many decoding algorithms used today are based on linear models, and the features used for lower

limb decoding remain at the early stage where simple filtered frequency bands are used for decoding [40, 41, 42, 43]. Recent studies in our lab showed an implementation of non-linear real-time decoding using an unscented Kalman filter (UKF) with delta-band EEG as a feature of neural activity [30, 44]. Although this improved decoding performance, it raised several questions to be explored as described below.

Previous research has shown the feasibility of using EEG to decode joint angle kinematics [42, 45, 46]. Presacco et al. 2011 [42] demonstrated that the decoding performance of joint angles from EEG was comparable to those using multiple single-unit activities recorded in nonhuman primates. They identified the optimal number of electrodes for the decoder and observed that the fronto-posterior cortical networks were heavily involved in gait. Luu et al. 2015 [46] showed the feasibility of using a closed-loop BCI to control a walking avatar under both normal and altered walking conditions while participants were introduced to visuomotor perturbations involving cortical adaptations. Luu et al. 2016 also demonstrated the use of a non-linear neural decoder using an unscented Kalman filter to decode joint angles during human treadmill walking using delta-band EEG as the predictor [45]. The decoder they developed was robust to ocular artifacts and allowed for real-time implementation. Similarly, Hikaru et al. 2019 developed a decoder to estimate muscle synergies and individual muscle activation from delta-band EEG as well, revealing the cortical correlates of muscle synergy activation associated with location [47, 48].

However, these studies have certain limitations that we intend to address in this paper. Most of the prior studies have used specific tap sizes for estimating the instantaneous joint angles without comparison. Additionally, none of the studies explored the ideal tap size for continuous decoding within the real-time implementation. Even though most of these studies claim high decoding performance based on r-value, a higher r-value does not always imply perfect tracking. Especially since a prediction which follows the general trend, but is way off from the actual values would still

contain a high r-value. Also, no studies have compared the performance of using different models and/or filters for continuous decoding, and all the above-cited studies made use of only delta-band power for joint angle prediction. Therefore in this study, we also explore the role that other frequency band bands may have on decoding performance.

We performed the experiments to prove the above mentioned points using online equivalent preprocessing and offline decoding combinations. It is true that online preprocessing and online decoding combinations are more idealistic, but to rigorously test a wide variety of combinations of parameters and decoders to compare against each other, we chose this approach. To make our approach feasible, we included the same preprocessing and decoder combination we have previously used in a real-time closed-loop experiment [44, 45] and treated the data in a similar manner.

The overall goal of this paper is to investigate what kind of machine learning algorithms, under what condition, perform best for EEG-based gait decoding. Although online and offline decoding is different schemes, we believe one of the advantages of offline decoding analysis is to rigorously test different conditions in order to provide feasible options in the later design of an online decoder. In this context, we investigated how the following factors affect decoding performance: 1) algorithms, including the number of hidden units, 2) tap sizes, 3) downsampling effects, and 4) frequency band features. To address the above issues, we designed and conducted offline experiments rigorously comparing performance against each other to validate the aforementioned factors.

2.3 Materials and Methods

2.3.1 Data

The data set consisted of EEG and kinematics data from 8 healthy subjects with each subject undergoing three trials that were spread across two days. Each subject walked on a treadmill for a total of 20 minutes per trial. Each trial consisted of three different tasks: resting, (based on kinematic measurement using goniometers, "Gonio") Gonio control, and closed-loop BCI control (hereafter BCI control). Two minutes of baseline period where subjects were instructed to stand still on a treadmill was collected in each trial before and after the treadmill walking. In the beginning part of the experiment where subjects finished baseline period, subjects were instructed to walk on a treadmill at a fixed slow speed of 1 mile per hour (mph) and staring at the screen in front of them at the same time where real-time feedback of a virtual reality avatar was provided. The virtual reality avatar's lower limb movements were synced with the goniometers attached to the hip, knee, and ankle of the participants. During the task, subjects were instructed to walk steadily for 15 mins where the decoder is calibrated for the next BCI control. Following the Gonio control, the experiment switches to a BCI control where the right leg of the virtual reality avatar is now controlled using EEG to give real-time feedback to the subject. This phase of the trial continued for five mins. A 64-channel active EEG electrode system from BrainVision was used out of which 4 channels were used as electrooculogram (EOG) sensors to capture and remove eye related artifacts using adaptive filtering algorithms [49]. The sampling frequency was set to 100 Hz. See Figure 2.1 for an overview. The data set was collected in our previous experiments and is publicly available with a full description [50].

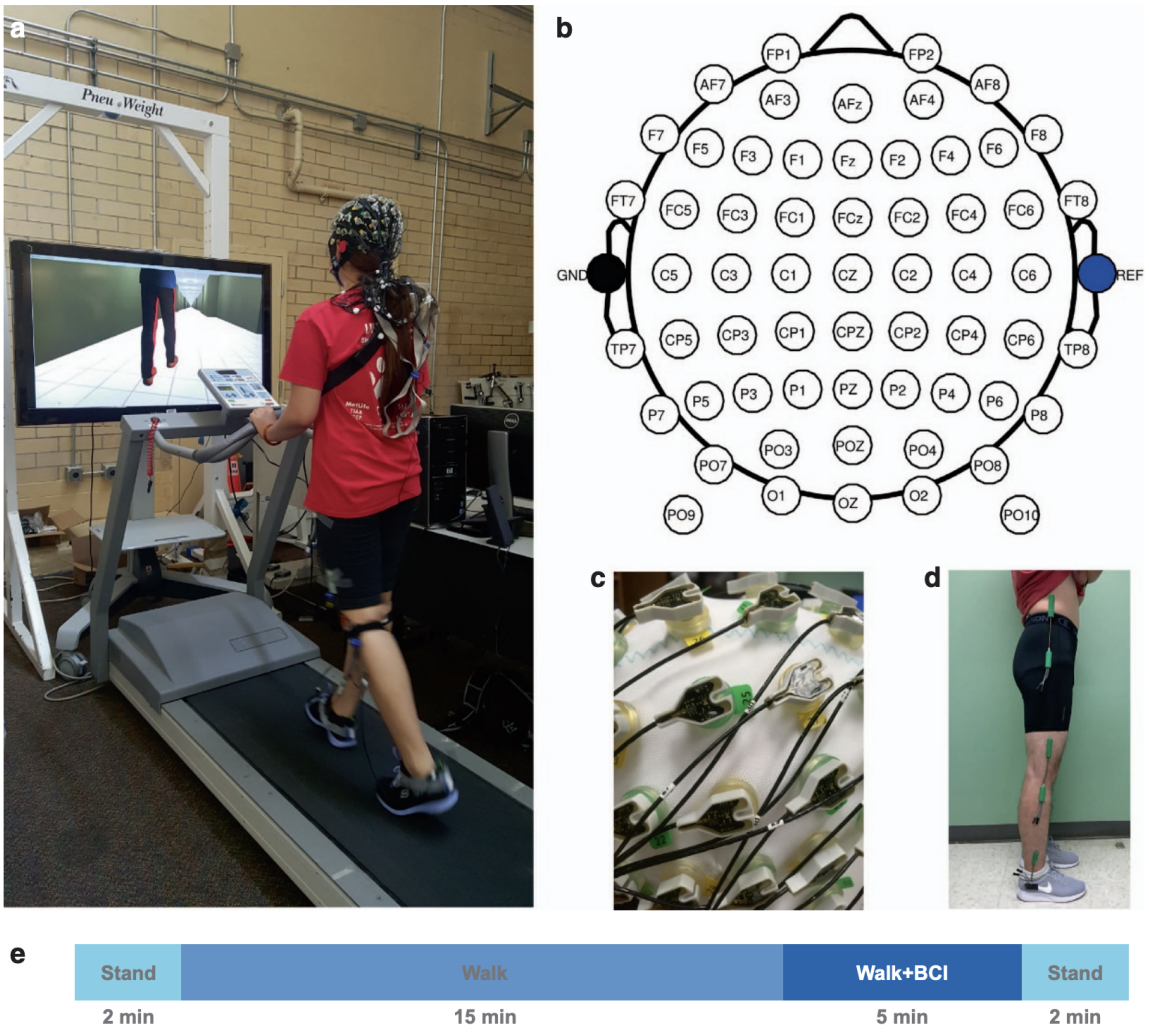

Figure 2.1: Experimental setup (a) Setup. (b) Montage of 60 EEG channels. (c) Active EEG/EOG electrodes. (d) A goniometer unit, consisting of two endblocks connected by a spring. (e) Protocol timeline. [50] CC BY 4.0).

Train, validation, test split

The data set was split into "train", "validation", and "test" sets in a sequential manner to simulate the online decoding scheme. For real-time decoding, each trial consisted of two modalities: Gonio control and BCI-control[44] (As described in the above Data section). Similarly, in our offline experiments, we utilized Gonio control to be the "train" and "validation" sets, where the first 80% was used as the "train" set

and the last 20% was used as the "validation" set. The Gonio control section had 60 channels by 15 mins x 60 seconds x 100 Hz = 90,000 time samples. The BCI control phase followed the Gonio control section. This section utilized the decoder trained during the Gonio control phase and used the model to decode the right leg in real-time using the EEG signals. This section had 5 mins of data and we used this entire section as the "test" set. The BCI control had 60 channels by 5 mins x 60 seconds x 100 Hz = 30,000 time samples. In our offline experiments, the "train" data set was used to train the model with certain hyperparameters, the "validation" set was used to assess the hyperparameter combinations used in the "train" data set. Finally, the "test" data set was used to assess the best hyperparameter combinations determined by the "validation" set. The example raster plot of three 10-second periods of data is shown in Figure 2.2.

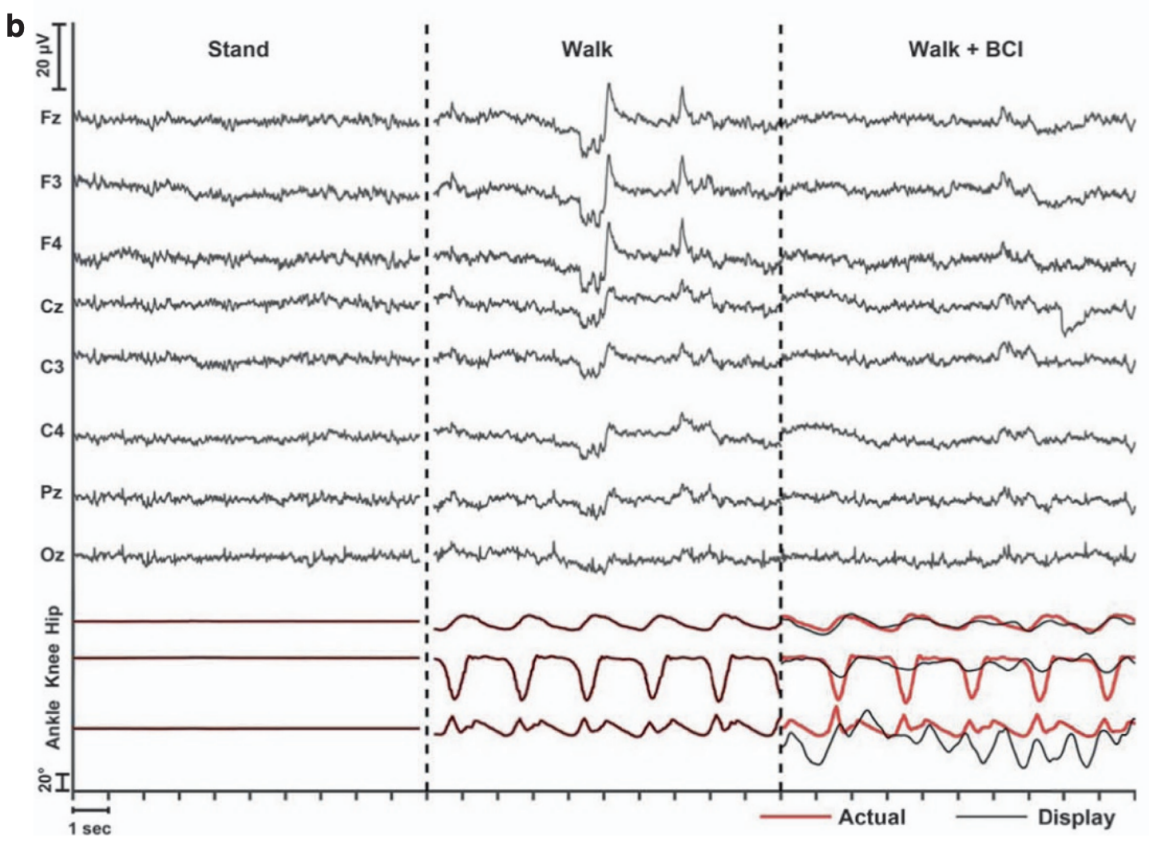

Figure 2.2: Sample raster plot of three 10-second periods. During the stand, walk, and walk+BCI phases (from folder SL04-T03). Adapted from [50] CC BY 4.0).

2.3.2 Code

The code is available on github: https://github.com/shonaka/EEG-neural-decoding. To replicate the environment, the Anaconda virtual environment in the github repository and the docker image are also available for replicating the building environment: https://hub.docker.com/r/snakagome/research_gpu.

2.3.3 Metrics

To quantify the decoding performance, two metrics were used: 1) Pearson's correlation coefficient (r-value) and 2) Coefficient of determination (R2 score). In the following equations, y is the actual joint angle and $\hat{y}$ is the predicted joint angle.

Pearson's correlation coefficient (r-value) was used in our previous studies to measure performance [44, 45].

$$\rho_{y,\,\hat{y}} = \frac{cov(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}}$$

where $cov(X, Y)$ is the covariance between the two variables and $\sigma(X)$ is the standard deviation.

The Coefficient of determination (R2 score) is another statistical metric used to measure the degree of variation of one data series that can be predicted from another. The formulation for R2 score is not to be confused with the squared Pearson's correlation coefficient. The value can be negative if the model overfits the training set and accounts for the variance accounted for by the model. Generally, r-value (Pearson's correlation value) would be a useful metric if the overall trends of the prediction with respect to the ground truth are of interest. However, if you want to quantify more precise errors between the two variables (prediction vs ground truth), the R2 score is a more suitable measurement. The R2 score was also used to evaluate similar decoding tasks using invasive data in the previous studies [31, 51].

$$R^2 \equiv 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual joint angle and y_i is the actual joint angle at time sample i.

2.3.4 Pre-processing and experimental designs

Pre-processing pipelines for different offline experiments are represented in Figure 2.3. The base pipeline is selected such that they can easily be used in an online real-time decoding scheme [44]. An H-infinity algorithm was used to specifically remove eye blinks, eye motions, amplitude drifts, and recording biases simultaneously [49]. The parameters of the H-infinity algorithms were kept the same as the real-time decoding. Peripheral channels were removed as they typically contain many artifactual components. The signals were then bandpass filtered using a 4th order Butterworth filter. Although the frequency range was the same, this is one of the differences compared to the real-time decoding as the real-time implementation utilized finite impulse filter and the phase shift was expected. To this point, all processing was done through a MATLAB script, which is also provided in the open-sourced repository. Additionally, before each experiment, the signals were z-scored for each channel.

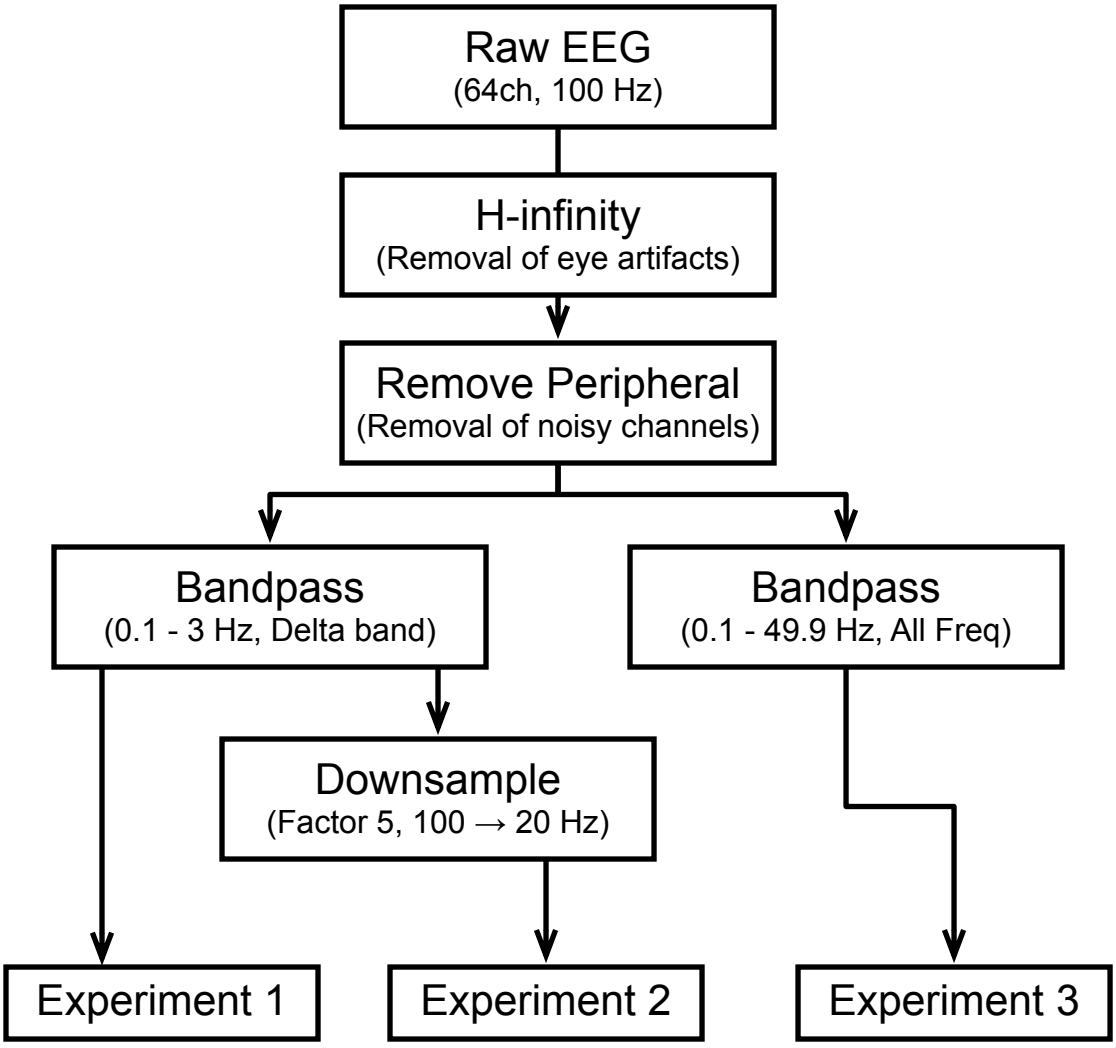

Figure 2.3: A preprocessing pipeline for two experiments. Experiment 1 is equivalent to [44]. Experiment 2 is downsampled. Experiment 3 contained all the frequency bands. Adapted from [10] CC BY 4.0).

Experiment 1: Decoding based on delta band features

The protocol for Experiment 1 is equivalent to the real-time decoding pipeline used in the previous studies [44, 45]. This is the baseline data processing pipeline, which will be used as a comparison for the following two experiments. We first calculated the performance metrics for each trial. We then calculated the median value for each tap size for each algorithm to draw a marker for visualization on figures. The error

bars were also calculated and plotted using 25th to 75th percentile range.

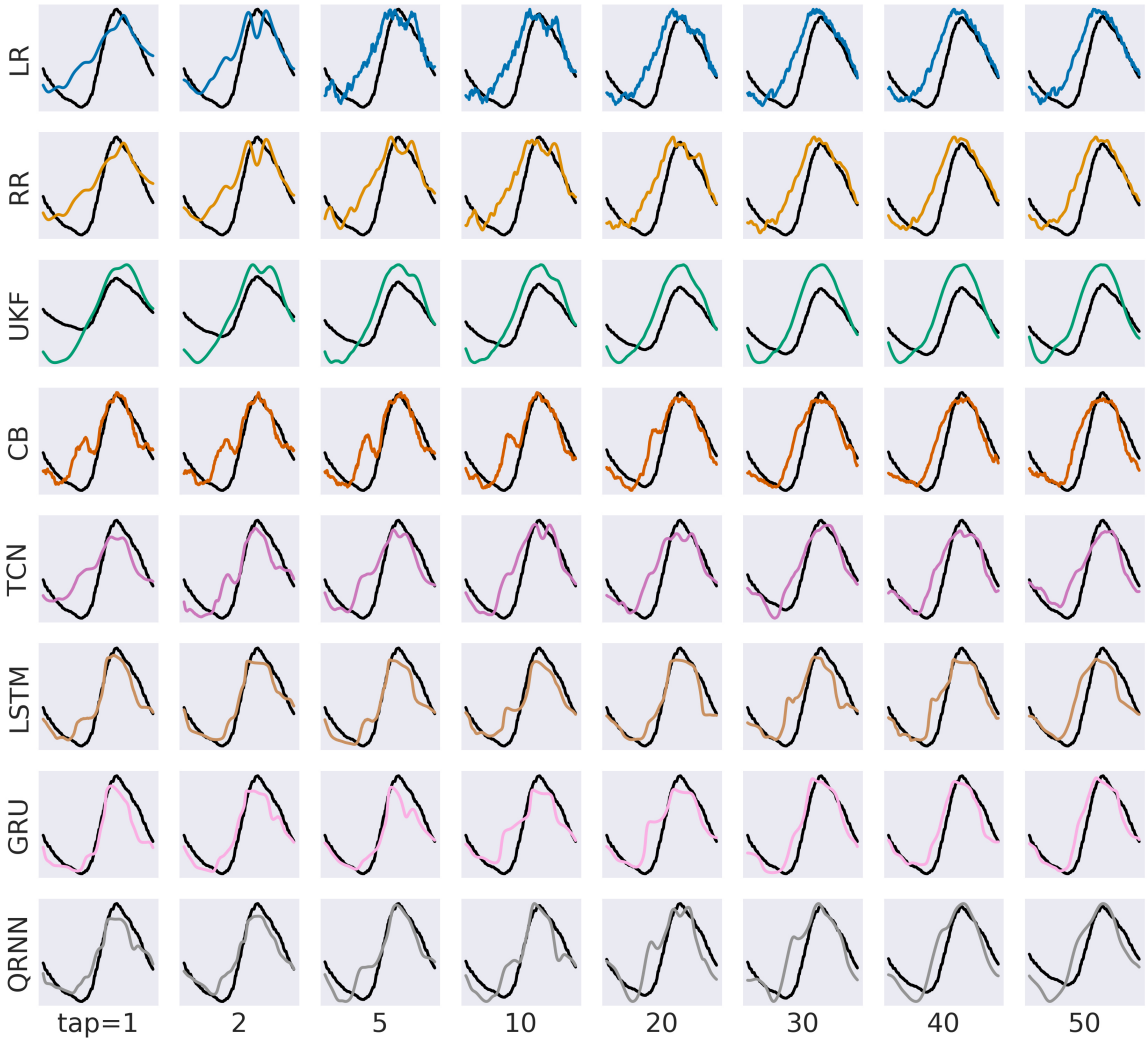

Figure 2.4: Example decoding results for hip joint for one gait cycle. Adapted from [10] CC BY 4.0).

Experiment 2: Downsampling effect

The primary goal of Experiment 2 was to investigate the effect of downsampling on delta band band-passed time samples. EEG data were resampled from 100 Hz to 20 Hz. Similar to Experiment 1, the median performance was calculated across all trials. To see the difference in performance as compared to Experiment 1, the performance of Experiment 1 was subtracted from the performance of Experiment

2. The black line for zero was added to see which Experiment performed better at certain tap sizes.

Experiment 3: Other frequency bands

The primary goal of the final experiment 3 was to investigate the effect of using all the frequency bands as opposed to just using the delta band features. We utilized the same bandpass filtering parameters except with a modified frequency range (0.1 - 49.9 Hz). As with Experiment 2, a similar analysis was performed to assess the effect of automatic feature learning from different frequency bands.

Tap sizes

Tap size refers to the number of samples in history used to train the model. A decoding schematic explaining the concept of decoding using a sliding window is presented in Figure 1.1. The figure shows an example where the tap size is five. The tap size of five was also the tap size we used in real-time decoding to collect the data[44]. In this paper, to thoroughly test the effect of tap sizes, we tested the model with different tap sizes: 1, 2, 5, 10, 20, 30, 40, 50. Given that our sampling frequency was 100 Hz, this is equivalent to using a tap size of 10, 20, 50, 100, 200, 300, 400, 500 ms of past data to predict the 10 ms future. This was common in both Experiments 1 and 3. On the other hand, in Experiment 2, we only utilized tap sizes until 20. Considering the fact that the downsampled frequency is now 20 Hz, the tap sizes in the downsampled scenario correspond to 50, 100, 250, 500, 1000 ms for 1, 2, 5, 10, 20 tap sizes, respectively.

2.3.5 Algorithms

The following eight algorithms were compared against each other: Linear regression (LR), Ridge regression (RR), Unscented Kalman Filter (UKF), CatBoost (CB), Temporal Convolutional Network (TCN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Quasi Recurrent Neural Network (QRNN). Within each architecture, the hyperparameters were listed below in each algorithm section. The hyperparameters were optimized using Bayesian optimization, which will be described in detail in the next subsection.

Linear Regression (LR)

Linear Regression is one of the most basic machine learning methods typically used to model the predictive relationship between the dependent target variable to multiple explanatory variables[52]. Note that in this context, Wiener Filter is equivalent to LR because of the way we feed in the input as a time sequence manner. However, in the machine learning context, we are denoting this as LR.

Ridge Regression (RR)

Ridge Regression is linear regression with L2 regularization [53]. RR is equivalent to WienerRR in this context. The parameter optimized during the training was α, which determines the strength of the regularization. It performs regularization so that the features that influence the target-dependent variable the least get penalized the most.

Unscented Kalman Filter (UKF)

Unscented Kalman Filter is an improved version of Kalman Filter [54]. It utilized an unscented transform to incorporate non-linearity within the model. In this specific context, we are following the implementation from Li et al. where UKF was first used to decode the kinematic movements invasively with monkeys[6]. The parameters optimized here are λ_F, λ_B, and κ.

CatBoost (CB)

Catboost is one of the most recent gradient boosting algorithms over decision trees[55]. The parameters optimized here are the learning rate, depth, and L2 regularization term. This was initially employed to compare against other gradient boosting algorithms, so we only picked the parameters common to these algorithms. Also, even with GPU capability, gradient boosting optimizations take a long time. There is still room for optimizing this algorithm given other parameters that we did not optimize.

Although we also implemented XGBoost[56] and LightGBM[57], we did not observe much of the performance difference between the gradient boosting algorithms. Since catboost was the fastest when computing using a GPU, we decided to remove both XGBoost and LightGBM from further analysis. However, the implementation is readily available on the Github link.

Temporal Convolutional Network (TCN)

Temporal Convolutional Network (TCN) is a specific type of convolutional neural network (CNN) architecture where a dilated causal convolution is wrapped with a residual block[58]. The authors compared the performance against other well known RNN architectures such as ordinally RNN, LSTM, and GRU and showed superior

performance across all tasks [58].

For all the neural network types of architectures from here below, there were common parameters that were optimized. First, the optimizer was optimized among ADAM, Stochastic Gradient Descent (SGD) with momentum, and AdaBound[59]. The learning rate and weight decay were also optimized. In addition, the number of epochs was also optimized as this is subject to change with the other parameters such as learning rate and weight decay. Specifically for TCN, the number of filters, layers, and kernel size were also optimized.

Long Short Term Memory (LSTM)

Long Short Term Memory is a sophisticated version of recurrent neural networks (RNN) where three gates are added to control the information to retain and pass [60], while avoiding the problem of vanishing gradient typically associated with training of a regular RNN.

In addition to the common parameters in the TCN, recurrent neural networks (LSTM, GRU, QRNN) had a number of hidden units, layers, the standard deviation for layer initialization, and clipping strength (which helps to prevent the gradient from exploding) were optimized. We did not observe the gradient exploding in TCN so this was omitted from the TCN optimization.

Gated Recurrent Unit (GRU)

Gated Recurrent Unit is another improvement to the RNN where it has two gates to control how to retain and pass information between the nodes. The same parameters were optimized as LSTM [61]. Even though previous empirical evaluations[62] have not shown a clear winner between GRU and LSTM, it is speculated that GRU could be a better model when dealing with a lower number of data to generalize upon,

considering the fewer number of parameters in comparison to LSTM.

Quasi Recurrent Neural Network (QRNN)

Quasi Recurrent Neural Networks (QRNN) is another alternative to a normal RNN where computations can be performed in parallel rather than sequential using convolutional layers [63]. The sequential dependencies in QRNN are handled using pooling, which makes the algorithm efficient to compute. The original paper that proposed the method showed its superior performance when compared against LSTM in a language modeling task. As for the actual implementation of the QRNN, we utilized QRNN implementation in fastai library[64]. The same parameters were optimizied as with the LSTM.

2.3.6 Hyperparameter optimizations

A Bayesian optimization library called Optuna[65] was used in this study. The number of trials was set to the default of 100 except for RR, where only one parameter had to be tuned (trials = 50). Optuna also provides an automatic early stopping for unpromising trials to save time, which is called pruning. For pruning, an asynchronous successive halving algorithm was used with default parameters. In all the optimizations, the mean squared error was chosen as the metric to be minimized.

2.3.7 Post-Analysis

After testing the model with test data, the following post-analysis was performed to investigate the patterns of preference of the two best decoding algorithms and the feature of importance in all the algorithms.

Determination of the optimal number of layers and hidden units

To investigate the patterns of performance for the two best algorithms (GRU and QRNN), we conducted a grid search fixed number of layers and hidden units analysis for a tap size equal to five (equivalent to the real-time implementation). We optimized for the other parameters with the fixed number of layers and hidden units using optuna with 100 trials. 100 combinations were created where the number of layers differed from one to ten layers and the number of hidden units ranged from 8 to 80 with an incrementation of 8 for each step (8, 16, 24, 32, 40, 48, 56, 64, 72, 80). This way, each combination is optimized for the highest performance for the specific combination of the number of layers and hidden units. This is to make sure to exclude the possibility that certain combinations of the number of layers and hidden units favor certain hyperparameters.

After computing the 100 combinations of the number of layers and hidden units with 100 trials of hyperparameter optimizations for each combination, we plotted the average performance for each combination in a grid heatmap for each metric to study the pattern or the tendency of GRU and QRNN.

Feature of importance

To fairly compare the decoding models with a single feature of importance model, we utilized channel-by-channel input perturbation on a trained model. For this, we utilized the model with the tap size = 5 because 1) this was the particular tap size we used in real-time decoding, 2) to shorten the time (the analysis takes a long time even for this single tap size as it has to perform one testing per channel and there were 46 channels). We fed the data with a single channel randomly permutated in the time samples. The random seeds were fixed throughout the experiments for replicability.

The performance was assessed after testing (making predictions) with the single-channel perturbation and the same process was repeated for all the channels one-by-one. By subtracting the ground truth where all the channels were not perturbed as we performed for the assessment of Experiment 1, we could evaluate the feature of importance in channels where the performance significantly drops. We also sorted the channels in order so as to pick the top five channels where the performance decreased the most, which indicates the importance of the channel.

2.4 Results

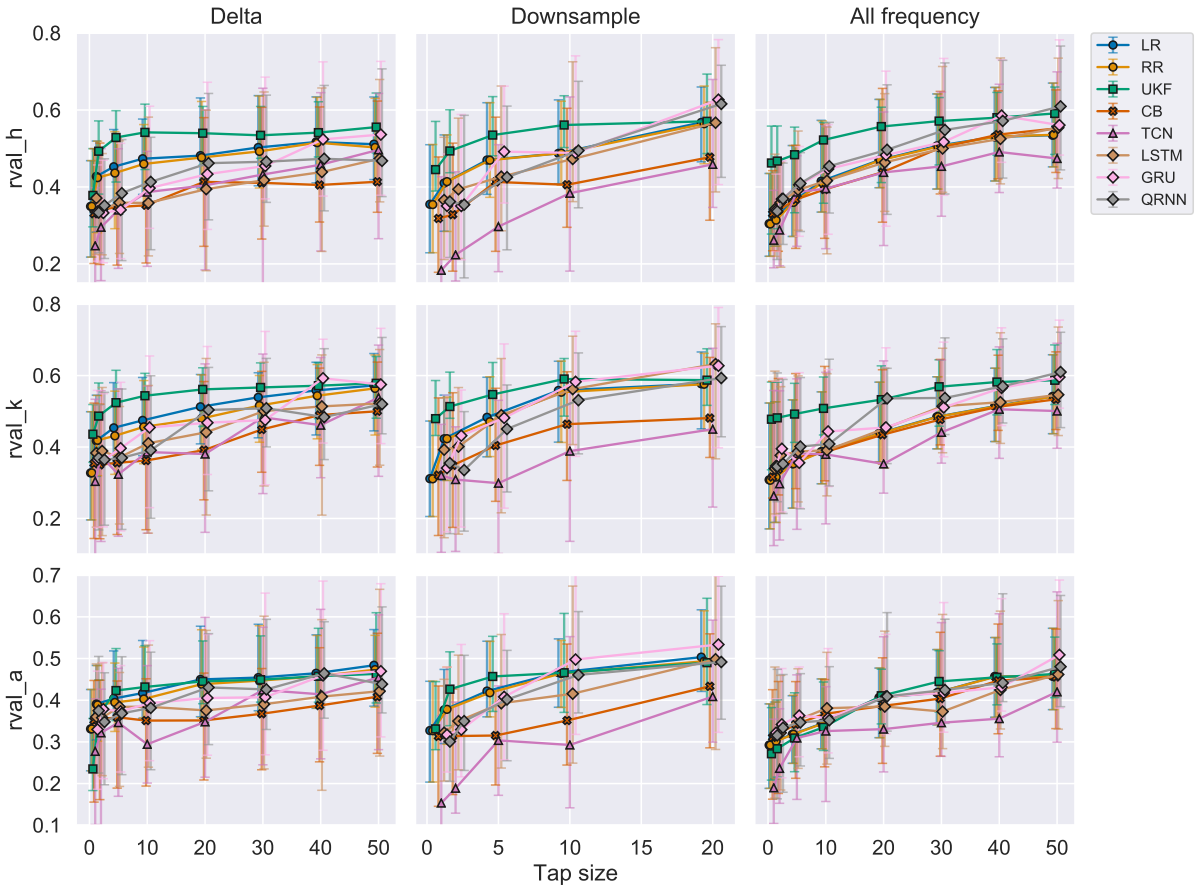

Figure 2.5: R-value evaluation. Each row represents each joint angle and column represents experiments. Each marker shows the median performance, errorbar shows 25th to 75th percentile. Similar algorithms were grouped using shapes. Adapted from [10] CC BY 4.0).

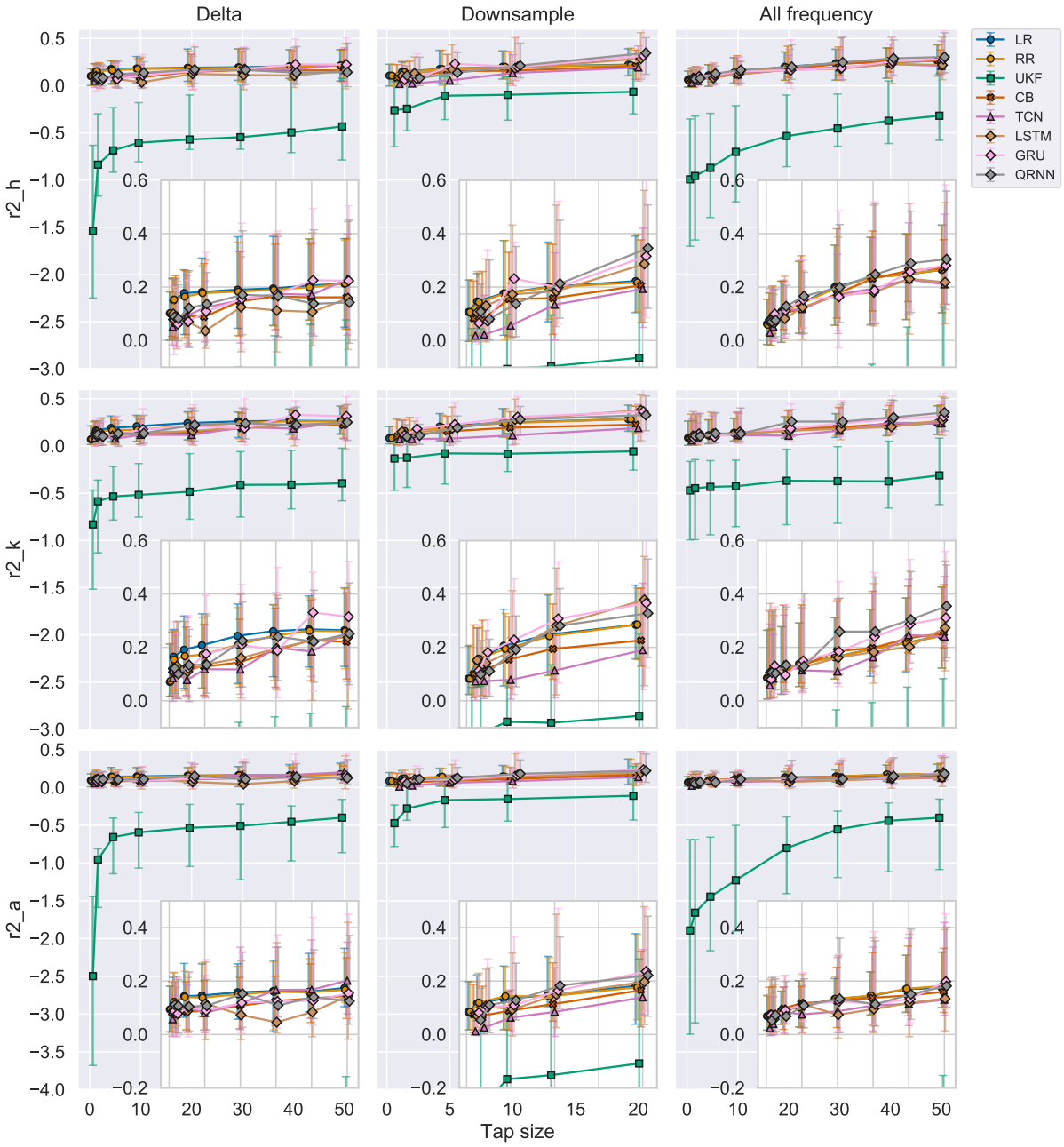

Figure 2.6: R2 scores. Each row represents each joint angle and each column shows each experiment. Each marker is a median and errorbar shows 25th to 75th percentile. The inset figures are the magnified version to compare algorithms other than the UKF. Adapted from [10] CC BY 4.0).

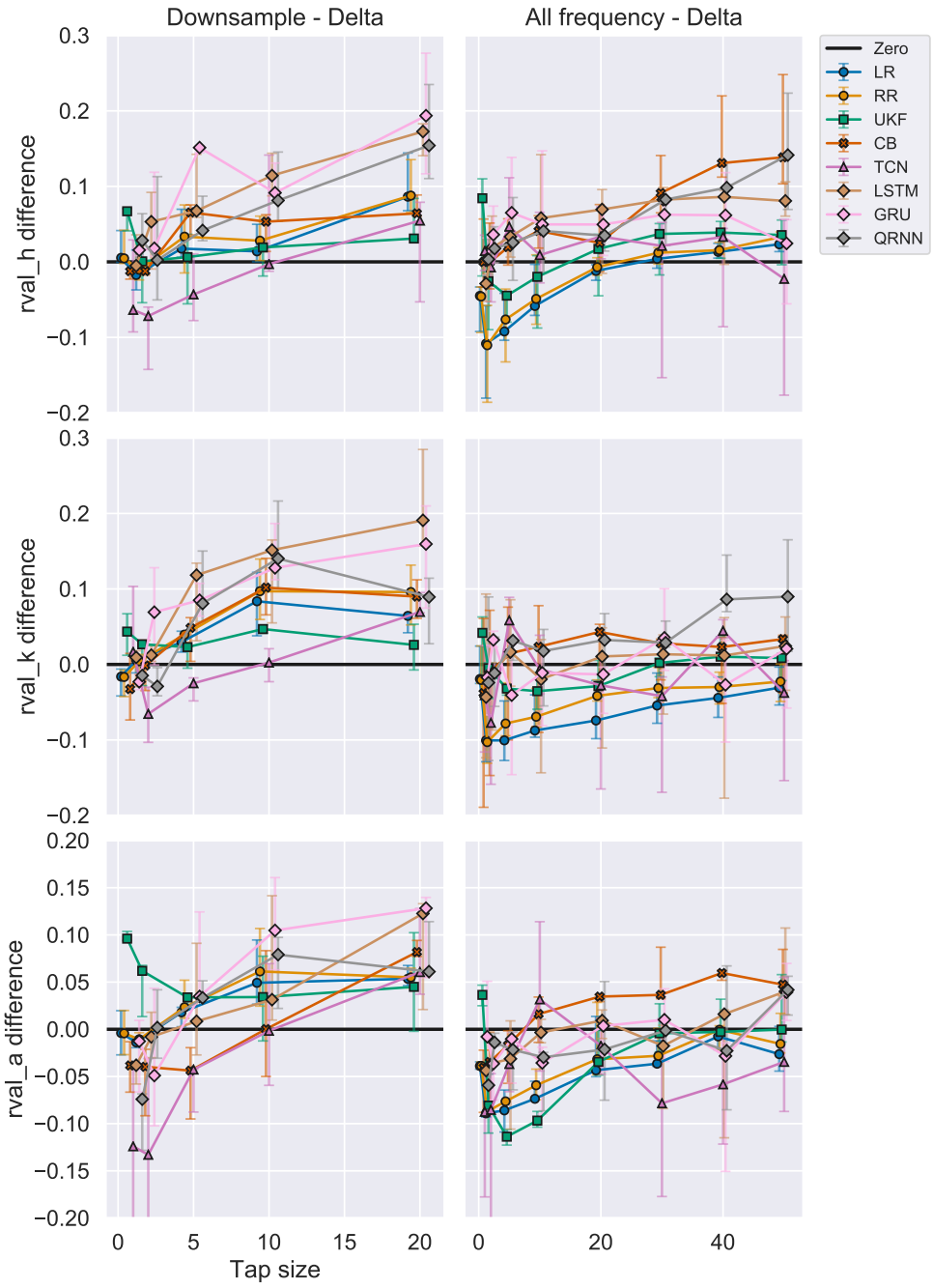

Figure 2.7: R-values difference between 1) Downsample vs Delta and 2) All frequency vs Delta. The black line shows a zero threshold to indicate which experiment performed better. Adapted from [10] CC BY 4.0).

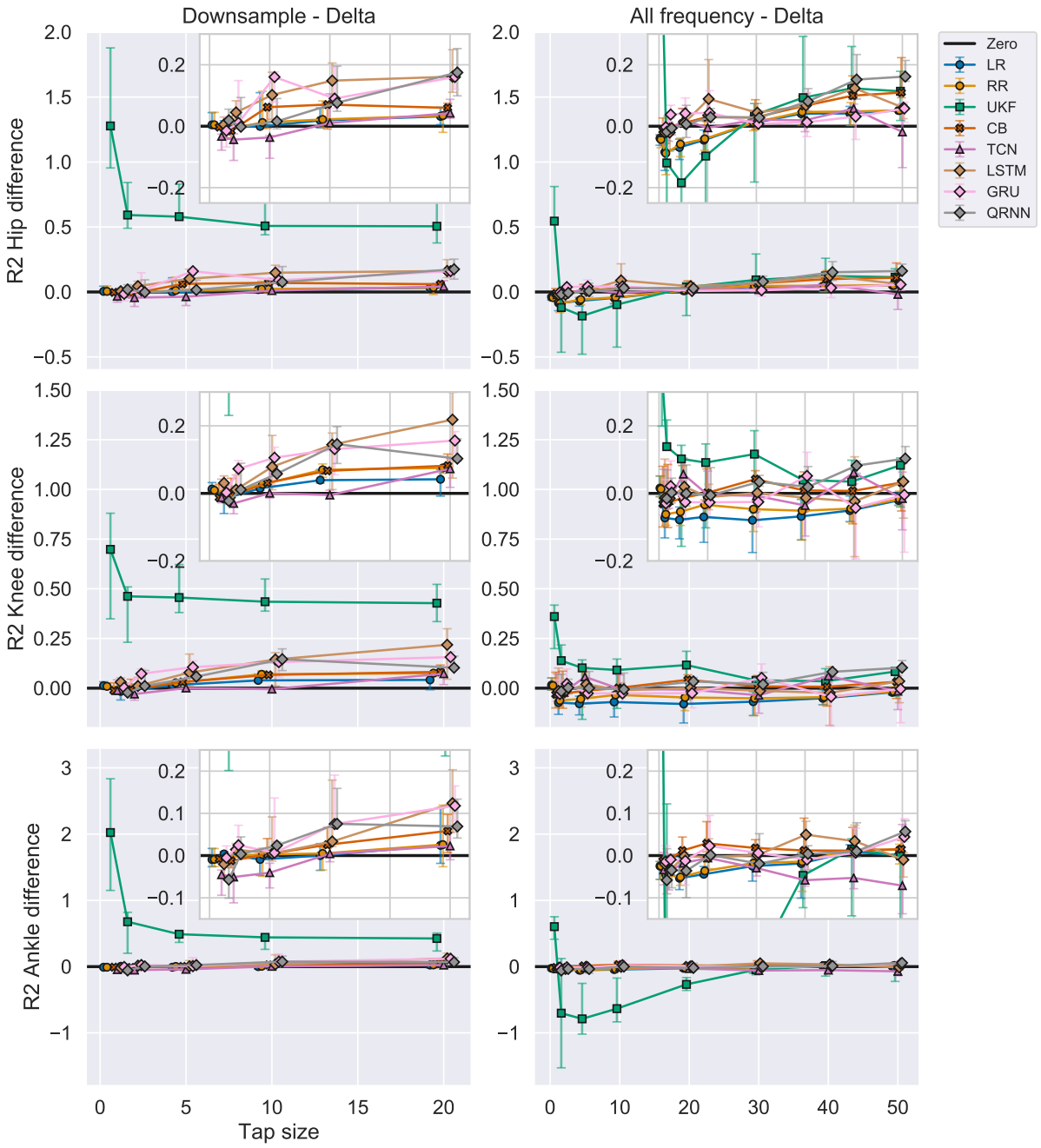

Figure 2.8: R2 difference with UKF between 1) Downsample vs Delta and 2) All frequency vs Delta. The black line shows a zero threshold to indicate which experiment performed better. Adapted from [10] CC BY 4.0).

2.4.1 Experiment 1: A comparison of different algorithms

To assess how different algorithms perform with the same pre-processing pipeline as the real-time EEG decoding, we performed a rigorous comparison based on variable tap sizes and quantified the performance based on r-values and R2 scores.

Figure 2.4 shows an example of decoding results of the best subject for each algorithm with different tap sizes. Each row represents different decoding algorithms in different colors. Each column represents different tap sizes. The predicted joint angles using different algorithms tend to become more smooth and close to the actual joint angles in the black line. When focused on each algorithm, we observed that the linear decoders (LR and RR) tend to be noisy when compared to other algorithms. UKF has the smoothest curve compared to all the other algorithms but tends to be off from the ground truth. GRU, the best algorithm among the compared algorithms, aligns well with the ground truth after the tap size of 30.

Evaluating from r-value perspective

Figure 2.5 shows a comparison of performances among each algorithm measured by r-value. Each row represents different joint angles and each column represents different experiments as described in Figure 2.3. In this section, we are specifically focusing on the first column. Each marker represents a median r-value across all the subjects and trials. The marker shapes represent similar algorithms with a circle denoting linear algorithms, square for Kalman filters, cross for boosting, triangle for CNN, and diamond for RNNs. All the algorithms tend to increase their performance as the number of tap sizes increased. This is apparent in the errorbar range of 25th to 75th percentile as the range also increased the performance. UKF showed superior performance across different tap lengths for the hip and knee joint reaching an average r-value of more than 0.50. LR and RR showed superior performance across different

tap lengths for the ankle joint reaching an average r-value of more than 0.45. On the other hand, CB, TCN, and LSTM performed worse in this metric.

UKF also reached 90% of the accuracy in r-value with respect to the maximum r-value across all the tested tap sizes when it reached the tap size equal to five. LR and QRNN also reached their 90% of the maximum accuracy after a tap size of 20. Other algorithms tend to require larger tap sizes as the accuracies continued to grow even after a tap size of 50.

Evaluating from R2 score perspective

Figure 2.6 shows a comparison of performances among each algorithm measured by R2 score with and without UKF, respectively. We are specifically focused on the first column for Experiment 1 where each marker represents a median R2 score across all the subjects and trials. Each row represents different joint angles. UKF significantly under performed as compared to other algorithms (Figure 2.6). We observed that the LR and RR outperform other algorithms with smaller tap sizes, but this was overcome by other algorithms such as GRU and TCN as tap size increased.

2.4.2 Experiment 2: Downsampling effect on decoding performance

Experiment 2 from Figure 2.3 was performed to investigate the effect of downsampling on performance. In the following sections, we assessed the performance using the two metrics and comparing against the baseline data processing pipeline in Experiment 1.

Evaluating from r-value perspective

Figure 2.5 in the second column shows the r-value performance for the Experiment 2. We still see the UKF dominates the performance in the smaller tap sizes up to 10 in hip and knee and 5 in the ankle, but GRU and QRNN start to perform better as tap size increased. The performance of CB and TCN in this experiment still comparatively underperformed, as in experiment 1.

To clearly see the difference in performance compared to Experiment 1, we subtracted the performance of Experiment 1 from the performance of Experiment 2, which is displayed in Figure 2.7 in the first column. The black line represents zero performance difference between the two experiments. We could see the performance in most of the algorithms increased with the downsampling except for the TCN in the smaller tap sizes. In fact, RNNs (LSTM, GRU, and QRNN) improved its performance more than 0.1 in r-value for hip and knee, and 0.05 in r-value for ankle joint. UKF had a unique trend where the performance increase was at its highest for tap size = 1 and slowly reduces to a plateau of performance at 0.025 as the tap sizes increased.

Evaluating from R2 score perspective

Figure 2.6 in the second column shows the R2 score for Experiment 2 with and without UKF, respectively. As represented in Figure 2.6, UKF performed the worst compared to all the other algorithms. To clearly show the other algorithms performance, we also included inset figures that magnify the other algorithms for comparison. In this experiment (2nd column), we could observe the RNNs (LSTM, GRU, QRNN) performed well compared to other algorithms across different tap sizes except for the smallest of tap sizes (1, 2, and 5) and certain joints (hip and ankle).

To evaluate the performance difference in R2 score compared to Experiment 1, we compared the R2 score by subtracting the performance of Experiment 1 from

the performance of Experiment 2 (Figure 2.8 in the second column). The black line represents the point of zero performance difference between the two experiments. UKF significantly increased its performance in Experiment 2 compared to Experiment 1 (Figure 2.8) although the improved performance evaluated in R2 score was still the worst as compared to other algorithms (Figure 2.6). The performance increase for UKF plateaued after the tap size of 2. Overall, RNNs (LSTM, GRU, and QRNN) significantly increased its performance in Experiment 2 compared to Experiment 1. The performance of CB also increased essentially consistent with an increase in tap size. Linear algorithms such as LR and RR showed a minimal increase in performance with downsampling.

2.4.3 Experiment 3: Automatic feature learning in neural networks

Experiment 3 (see Figure 2.3) investigated how frequency band features, other than the delta band, affect decoder performance. Similar to Experiment 2, we assessed the performance based on the two metrics.

Evaluating from r-value perspective

In the third column of Figure 2.5, we see the performance changes with different tap sizes for Experiment 3. The trend was very similar to Experiment 2, where UKF initially performed well for hip and knee joints where the tap sizes were small, but then GRU and QRNN outperforms the UKF. One difference compared to Experiment 2 is that, for the ankle joint, GRU and QRNN always performed better than the UKF in all the tap sizes.

To evaluate the performance difference compared to Experiment 1, we subtracted the performance of Experiment 1 from Experiment 3 and showed the difference in

Figure 2.7 in the second column. One interesting trend in the performance difference is that the linear decoders (LR and RR) and UKF tend to perform worse when all the frequency was used. This effect is minimized as the tap size increased. On the other hand, boosting algorithms (CB) and neural networks gained a slight increase in performance (around 0.05 in r-value).

Evaluating from R2 score perspective

Figure 2.6 in the third column shows the performance evaluated for the 3rd experiment using R2 score. Again, UKF did not perform well when evaluated from R2 score. In the third column of the same Figure, QRNN performed well across different tap sizes for hip and knee joints. Ankle joints showed similar performance among different algorithms where it was difficult to conclude which algorithms performed the best.

To evaluate the performance difference compared to Experiment 1, we subtracted the R2 score of Experiment 1 from Experiment 3 and showed the difference in Figure 2.8 in the third column. In Figure 2.8, UKF showed a significant increase in performance in tap size = 1, but the Experiment 3 with all the frequency band features performed well in tap sizes 2, 5, 10 for hip and 2, 5, 10, 20, 30, for ankle joint. The only exception was the knee joint where the performance in Experiment 3 for UKF was still superior compared to the R2 score in Experiment 1.

Figure2.8 showed an interesting pattern where the performances of linear algorithms such as LR and RR did not increase in Experiment 3 as compared to Experiment 1 in smaller tap sizes (1, 2, 5) for hip, all the tap sizes except 1 in the knee, and tap sizes until 30 in the ankle joint. CB constantly showed a performance increase across different tap sizes and joints. The performance of TCN did not increase for the hip and knee joints and decreased for the ankle joint. RNNs (LSTM, GRU, QRNN) showed the largest performance increase in the hip joint, but only some of the RNN

algorithms had a performance increase in larger tap sizes for knee and ankle joint angles.

2.4.4 Different number of layers and hidden units

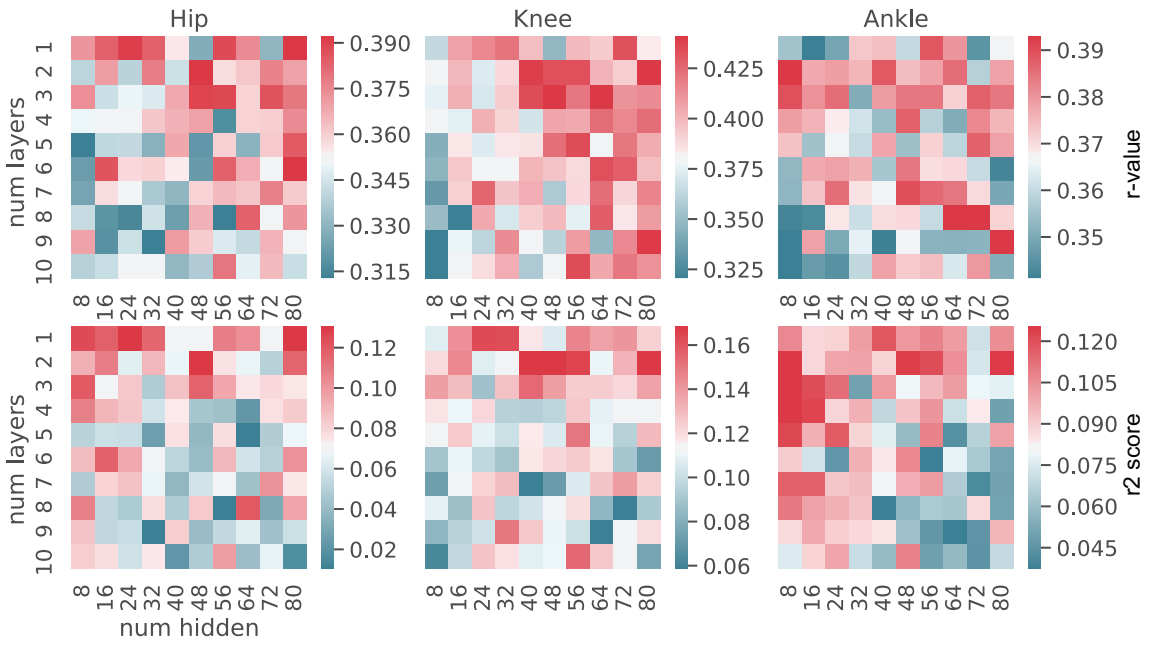

Figure 2.9: GRU assessing number of layers and hidden units patterns. The row shows median r-value and R2 score for each joint. The color bar indicates the median performance across trials and subjects. Adapted from [10] CC BY 4.0).

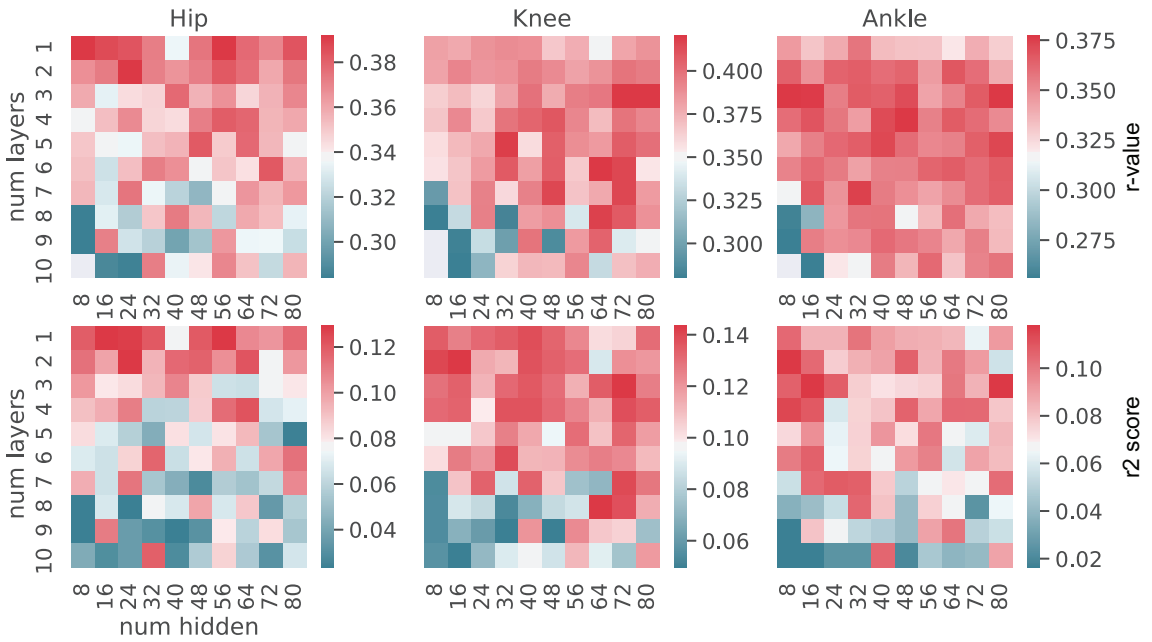

Figure 2.10: QRNN assessing number of layers and hidden units patterns. The row shows median r-value and R2 score for each joint. The color bar indicates the median performance across trials and subjects. Adapted from [10] CC BY 4.0).

For the two well-performing neural networks (GRU and QRNN), we conducted rigorous experiments to investigate the patterns of the number of layers and hidden units. For each combination of layers and hidden units, we optimized for the hyperparameters to obtain the best performance for the combination. This way, other parameters would not bias or favor some particular combination of layers and hidden units. We calculated the median performance across trials for the combination and represented them as a heatmap for each metric and each joint (Figure 2.9 and 2.10). GRU (Figure 2.9) and QRNN (Figure 2.10) showed similar patterns. Overall, the performance was superior with a lower number of layers (1-5) compared to a higher number of layers (6-10). Patterns based on the number of hidden units differed between the metrics and joints. When the performance was evaluated from R2 score, general patterns suggested that a fewer number of layers and hidden units (8-40) showed better performance when compared to a higher number of layers and hidden

units.

2.4.5 Feature of Importance

Feature of importance in channel analysis revealed not only the importance of some EEG channels, but the robustness of the model. Figure 2.11 and 2.12 shows the results of feature of importance evaluated in the two metrics. Figure 2.11 presents the importance of channels evaluated based on the decrease in r-value with respect to the ground truth where no perturbation was performed. The x-axis shows the identified top five important channels from each algorithm and the y-axis shows the decrease in r-value when the channel was perturbed. Similarly, Figure 2.12 evaluated the feature of importance using R2 score decrease. There are four plots in this case because some algorithms significantly decreased its performance compared to other algorithms so that it was difficult to assess all the algorithms together in a single plot. All of the algorithms are represented in the 1st plot (1st row), the second plot excludes the linear models, such as LR and RR (2nd row), the third plot excludes the linear models and UKF (3rd row), and the final plot excludes linear models, Kalman filters, and TCN (4th row).

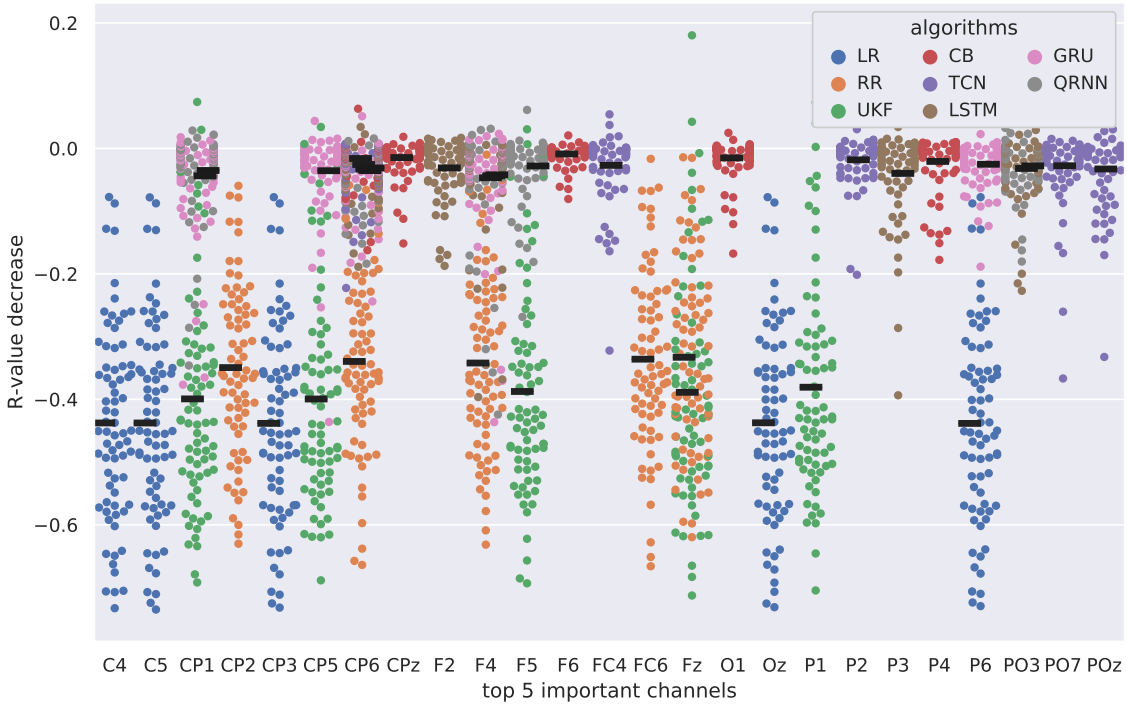

Figure 2.11: Feature of importance in channel assessment for each decoding algorithm evaluated with r-value. The thick black line represents the median decrease in r-value for each algorithm for each channel. Adapted from [10] CC BY 4.0).

Figure 2.12: Feature of importance in channel assessment with R2 score. The first plot includes all the decoding algorithms. The second plot without LR and RR. The third plot without LR, RR, and UKF. The fourth plot without LR, RR, UKF, and TCN. Adapted from [10] CC BY 4.0).

2.5 Discussion

This study aimed to determine better algorithms for EEG gait decoding from four different aspects: 1) Algorithms, including the number of layers and hidden units, 2) Tap sizes, 3) Downsampling, and 4) Frequency band features. We computed the decoding accuracy and also assessed the EEG channel of importance and robustness of the decoder. The results identified that the former state-of-the-art algorithms

(UKF) can still be one of the best algorithms when evaluated from r-value perspective but performed the worst when evaluated from R2 score perspective and were also vulnerable to the perturbations compared to the neural networks based algorithms. RNN based algorithms such as GRU and QRNN performed well when assessed from multiple perspectives. These algorithms were also the most robust algorithms as the perturbations in channels did not deteriorate the performance much compared to the other compared algorithms. Linear algorithms such as LR and RR are still widely used algorithms and they have their advantages as performing as the baseline. However, the performance of such algorithms significantly decreases when some channels were perturbed and this shows their vulnerability as a model.

The current study aimed to improve the decoding accuracy and robustness for the lower limb decoding using EEG from algorithm perspectives. Accurate lower limb decoding is important for controlling exoskeletons and neuroprosthetics for better usability and systems. In the previous studies, we showed the average accuracy across subjects could not exceed the r-value of 0.5 [44]. With such low to mid accuracy, participants may not be able to engage in the virtual reality feedback and the neural decoding could not extend to more practical applications of lower limb movements such as controlling exoskeletons or neuroprosthetic legs where a high decoding accuracy is necessary for safety and better usability.

It is important to note that there are four different combinations of preprocessing and decoders to prove decoding performances. 1) Offline preprocessing and offline decoding, 2) Offline preprocessing and online decoding, 3) Online preprocessing and offline decoding, 4) Online preprocessing and online decoding. The current paper focuses on the third combination although the fourth combination is the most idealistic scenario. However, the fourth combination is both challenging and time-consuming because the real-time validation has to be validated in a real-time manner and thus not suitable for testing every combination of parameters and decoders. To make our

study on the third combination feasible, we utilized the same preprocessing pipeline and the decoder that we have previously used in a closed-loop experiment [44, 45] and compare the performance against it to validate each method we used. We also made sure not to use any future data and using the exact same data acquired during the closed-loop experiment for training and testing, simulating the real-time applicability in future studies.

Linear algorithms such as LR and RR showed a standard performance that is not necessarily superior to other decoder algorithms but can be used as a baseline decoder prior to closed-loop BCI studies. UKF was previously used both in invasive [6] and noninvasive [45] real-time applications to show its capability in neural decoding. Therefore, this can be considered as the state-of-the-art in a sample by sample decoding. In this study, UKF still showed its superior performance in early convergence with smaller tap sizes and when evaluated from the r-value perspective. On the other hand, UKF showed its vulnerability when evaluated from the R2 score and also when a channel is perturbed. Although CB did not show superior performance among the algorithms compared in this study, the uniqueness of CB is that the choice of a feature of importance was different from other algorithms. Therefore, CB itself could not be the main decoder, but when ensembling the models to aim for higher accuracy, this algorithm could be considered as one of the algorithms along with better performing algorithms. Despite the previous study [58] showing TCN could outperform RNNs, this was not the case in this study. One of the reasons may be the fact that the number of training samples fed into the models at a time was low considering that the final goal is to build a real-time application where the number of tap sizes is limited. This can be indicated from Figure2.5 where Experiment 2 (Downsampling) experiment showed TCN performing the worst. On the other hand, we could observe that the R2 score of TCN becomes the best among all the other algorithms in tap size with 50 for hip and ankle joint in Experiment 1 (Delta, Figure2.6). However, from figure 6, when we used all the frequency bands, TCN did not perform as well, even

for a tap size of 50. This could be explained by the influence of the filter size. If the filters are small, they might tend to learn local features; in our case, this would be high-frequency components. By including all frequencies, TCN will have more variance that it can explain in the high-frequency range and the majority of the filters might focus on these high-frequency components. Since deep learning models might tend to overfit and explain the maximal variance, it might weigh the higher frequencies more. This might lead to the model not focusing on the delta/ slower oscillatory components which would also have discriminatory information. On the other hand, when you force it to look into the lower frequency (using delta band alone in experiment 1), since there is no information in the higher frequencies, it will naturally look for features in the delta band and increasing the tap size would essentially increase the information present in these lower frequencies.

In the original paper, the dataset used to bench test involves data that is either highly correlated in space or contains high-frequency information. Therefore TCN should learn to capture this information with their smaller filter size and that could be the reason for having comparable results with recurrent networks. Since EEG has a lot of low frequency information, TCN might not put emphasis on this information when using lower filter size. Therefore more analysis needs to be done on filter width and how it would affect the features learned.

RNNs, especially GRU and QRNN showed superior performance from multiple perspectives. The most notable performance increase can be observed in Experiments 2 and 3 which we will discuss more details below. These algorithms also showed the robustness to channel perturbation (Figure 2.12).

Generally, all the algorithms increased their performance as the tap sizes increased (Figure2.5, 2.6) as expected. UKF quickly reached a plateau in performance reaching 90% of the maximum accuracy with the tap size equals five. Linear decoders such as LR and RR reaches 90% of its maximum accuracy with the tap sizes 20 - 30. Other

neural networks kept increasing its performance as far as 50 in Experiment 1 and 3 or 20 in Experiment 2. Therefore, linear decoders and UKF tend to perform better at lower tap sizes but with more tap sizes, other algorithms could outperform the linear and UKF decoders. This could be used as one of the baselines when determining the tap size to be used in real-time decoding.

Downsampling of delta bandpass features (Experiment 2) generally increased the performance compared to the equivalent tap size performance in Experiment 1 (Figure2.7, 2.8). Only exceptions were some of the small tap sizes (1 and 2) and algorithms such as TCN. From the r-value perspective, RNNs benefited from the downsampling increasing the r-values up to 0.2 for hip and knee, 0.1 for the ankle for the best case (Figure2.7). From R2 score perspective, UKF benefited the most increasing the R2 scores significantly in the smaller tap sizes and gradually plateauing its performance increase as the tap sizes reach 20. However, even with such increases, the R2 score was still the worst compared to other algorithms (Figure2.6).

Previously, how sampling frequency would affect the performance was unknown [9]. This study investigated this issue in a sample-by-sample decoding scheme and showed that the performance could increase. We could still record the data using the highest sampling frequency, but if we were to use delta bandpass features, we could technically reduce the sample size to 20 Hz provided that we give enough frequency range for a reconstruction. With this approach, we could technically also increase the future prediction time from 1 ms (when 100 Hz) to 5 ms (in 20 Hz) with the same decoding scheme. We also showed that the performance could actually improve.

Linear decoders and UKF decreased their performance in some cases when comparing the performance of the delta band features (Experiment 1) and all the frequency band features (Experiment 3) as represented in Figure 2.7, 2.8. We observed that the performance for the hip joint decreased until the tap size of 20, whereas the

knee and ankle joint constantly showed decrease in performance in both metrics (r-value: Figure2.7, R2 score: Figure2.8). UKF also showed performance decrease until tap size equals to 15 for hip, 30 for knee, and all along when evaluated from r-value (Figure2.7).Similarly, when evaluated from R2 score (Figure2.8, the tap size equals 20 for hip and 30 for ankle (except tap size equals to one in both metrics). On the other hand, this was not the case for other algorithms (except for TCN) where the performance increased up to 0.15 in r-values (Figure2.7) and R2 score (Figure2.8).

These results not only validate the importance of feature extraction using the delta band for the performance for linear decoders and UKF, but it also shows us the boosting algorithm (CB) and RNNs could benefit from other frequency band features because of their ability to extract meaningful features for the performance.

GRU and QRNN showed similar preferences in the number of stacked layers and hidden units. We observed certain patterns of combinations of the number of stacked layers and hidden units that gave better performance compared to other combinations. For example, GRU showed better performance with less number of stacked layers when evaluated from R2 score perspective (Figure 2.9). This is in line with the latest review [9] where shallow networks were observed with intra-subject studies.

Linear decoders (LR and RR) and UKF showed vulnerability to the input perturbations when evaluated from both r-value (Figure 2.11) and R2 score (Figure 2.12). On the other hand, neural networks and boosting algorithms did not deteriorate their performance as much as well as the boosting algorithm (CB). This can be also considered that the RNNs and CB are more robust and not relying heavily on certain channels. CB also showed an interesting pattern where the identified feature of importance channels was unique from others. This could be beneficial when creating an ensemble model with other better performing algorithms so that the end model could not only be more robust but also perform well.

There are some channels that were identified as important from multiple algorithms such as F4, CP1 and CP6 (Figure2.11 and 2.12). Although it is difficult to conclude from sensor level analysis compared to the source level analysis, these channels are located near the primary motor area at the center (CP1) and posterior parietal areas (CP6) that are known to be involved during the gait. Since the subject receives feedback based on the movement of the avatar, there might still be instances of adapting to this new paradigm of walking. This might be the reason why F4 was selected as an important feature by multiple algorithms as they are reported associated with coordinating motor movements and in adapting to the gait.

The real-time applicability of the model is an important part of the design of the neural decoder. Linear models and Kalman filters such as UKF were previously shown to be implementable by many studies. On the other hand, neural networks and boosting algorithms in neural decoding context had not shown its feasibility except for a few studies in the invasive decoding and thus needs to be investigated further [66, 67].

More specifically, Sussillo et al. used a variant of RNN called Multiplicative RNN with big data collected throughout invasive neural decoding of kinematics [66]. The performance was compared against the state-of-the-art kalman filter during the time and the RNN not only showed a superior performance but also robustness to noise which is in line with our results shown in this paper. Tseng et al. compared the performance of a wiener filer, kalman filter, UKF, and LSTM [67]. They also reported similar trends in performance increase and robustness. The current results in this paper also prove these points and for the first time in a non-invasive context.

The preprocessing pipeline used in our methods is focused on real-time applicability as mentioned before in this section. Therefore, we did not use some methods such as independent component analysis (ICA) for artifact removal. In addition, artifacts are carefully taken care of in our data processing pipeline where eye-related artifacts

are removed using H-infinity filter [49], motion-related artifacts are minimized by ensuring the use of elastic mesh on the electrodes during the data collection, slow walking speed (1 mph), and the removal of peripheral channels [44, 45] Muscle related artifacts were also dealt with in a similar manner. Although we do not claim the EEG data are completely free from the artifacts, we performed reasonable data collection and processing to minimize the effects.

As a summary and recommendation, if the purpose of the lower limb decoding requires precise control (e.g., controlling an exoskeleton), based on the high R2 score, GRU or QRNN would be recommended with shallow layers and the small number of hidden units to start with. The tap sizes could be chosen from 10 to 20 to start as that is where the performance started to plateau. Further research is required to validate the applicability of such neural networks in real-time when the training is involved. If the purpose of the decoding is to show weak trends following the ground truth, such as virtual reality feedback, UKF might be sufficient as it could provide high r-value against the lower limb movement and it has already been shown as a real-time application. In either case, simple linear models such as LR and RR could also be implemented and be used a baseline benchmark results to further improve the performance of other algorithms when tuning hyperparameters.

Chapter 3

Towards a generalized neural decoder for gait

3.1 Abstract

Neural decoding in Brain Computer Interface (BCI) usually requires retraining of a model prior to the experiment due to the changes in neural characteristics over time and across subjects. This is a significant setback because more time and effort are required to achieve reasonable decoding performance. Therefore, the generalization of a model is crucial. In this study, we studied intra- and inter- subject decoding on a treadmill and overground walking, where electroencephalography (EEG) and electromyography (EMG) data were used to decode the lower limb kinematics (KIN). Intra-subject decoding generalized the model for an unseen trial and showed feasible performance across all the decoding combinations. Inter-subject decoding generalized the model for an unseen subject and this did not show feasible performance in both treadmill and overground walking except for the EMG to KIN decoding. To improve on the inter-subject decoding, transfer learning was used and the performance not only significantly increased, but the variance in performance also decreased compared to intra- and inter- subject decoding, indicating stabilization in performance. These results indicate that to deal with subject variability in EEG, transfer learning is the key to generalized decoder for gait.

3.2 Introduction

EEG is known to be variable in its signal across subjects due to physiological differences between individuals [68]. Therefore, it is challenging to create a generalized model that can be applied across subjects [9]. This is problematic because the model has to be retrained every time it is used, requiring more time to run experiments. In fact, this is one of the main issues in lower limb studies [33]. Building a generalized neural decoder could significantly reduce the amount of time for training and running experiments.

Furthermore, if we could create a generalized model for neural decoding, the analysis on the model could cast an insight on general features of the representation rather than individual features dedicated to decoding. Although still in the early stages, with most of the analysis work performed on the visual cortex or visually related areas [24], a recent study found that a trained RNN contained hidden units with activations similar to that of the motor cortex [69]. These insights could help us build a better hypothesis through reverse engineering [25]. However, much work on how to approach and analyze these trained models is needed to perform this kind of research.

EEG gait decoding is still at an early stage in research. To date, the ability to generalize neural decoders across trials has not been investigated. While a more common approach to generalizing models has focused on intra- subject decoding (across trials decoding), there is a shift towards developing models that generalize across subjects (inter-subject)[9]. This could have implications for clinical applications, where obtaining training data from an individual with motor impairment may not be feasible. In this case, models that generalize across subjects could be trained on one population and applied to another group in a real-time BCI. We will start our investigation towards the generalization from here.

To tackle the problem of generalization in neural decoding across subjects, a recent study summarized that there are two approaches: 1) Subject-invariant and 2) Task-calibration approach [70]. The subject-invariant approach is typical in a machine learning field where, given enough data, one tries to create a universal pre-trained model that can be used for new unseen data. The task-calibration approach requires some data from the current subject to calibrate or fine-tune the model [70], which could be considered transfer learning.

Alternatively, the third approach is to break down the decoding process into physiological components. A recent study on speech decoding built a decoding model by combining the decoding of 1) the neural signal to articulatory system kinematics and 2) mouth kinematics to acoustics, instead of directly trying to decode acoustics from the neural signals [71]. Although the idea was simple, having the intermediate kinematics not only enhanced performance with limited data, but also made the decoder more robust across subjects because of less variability in kinematic representations compared to the neural signal variability across subjects [71].

The purpose of this study is to investigate the generalization of the neural decoder for gait from an electrophysiological perspective. First, we investigate the generalization from intra- and inter- subject perspectives. Next, instead of directly decoding the kinematics of the legs from the neural signals, we break down the decoding process into two parts: 1) EEG to EMG and 2) EMG to kinematics decoding (Hybrid model inspired by previous study[71]).

3.3 Materials and Methods

3.3.1 Data

The dataset consists of EEG, electromyography (EMG), and kinematics data from 10 healthy subjects with each subject undergoing 20 trials on the same day. A 64-channel active EEG electrode system from BrainVision was used with a sampling frequency of 1000 Hz. Four channels were used as EOG sensors to capture and remove eye related artifacts using adaptive filtering algorithms [49]. Surface EMG signals were recorded using active bipolar electrodes (fixed electrode distance of 20 mm) and recorded at 1000 Hz (SX230 sensors and DataLOG MWX8, Biometrics Ltd, Newport, UK). The kinematics were calculated based on the 17 wireless inertial measurement units (Xsens MVN, Xsens North America Inc., Culver City, CA). The overview of the setup is shown in Figure 3.1. An example of the raw data is shown in Figure 3.2.

Figure 3.1: Fully instrumented subject and experimental gait course. (a) Able-bodied subject setup. (b) EEG channel montage and EOG locations. (c) Close-up image of active EEG electrodes. (d) Experimental gait course. Adapted from [72] CC BY 4.0).

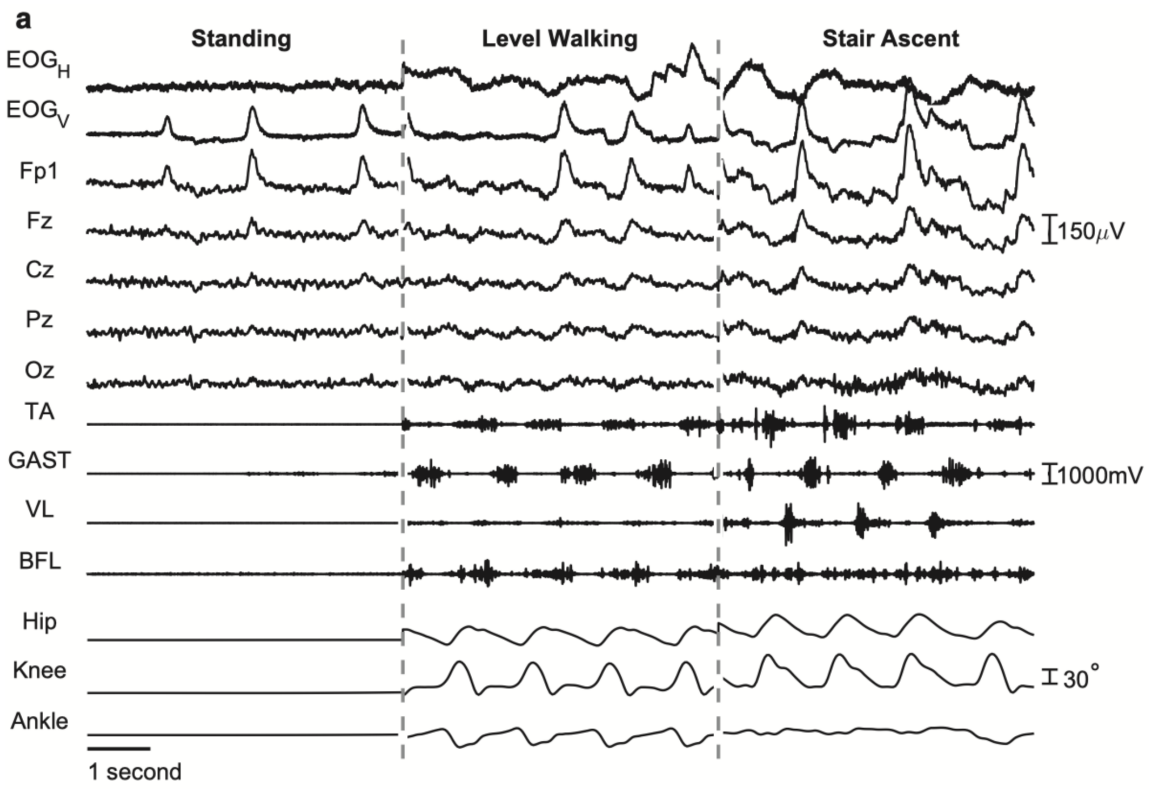

Figure 3.2: Example overground walking data in a raster plot. (a) The timeseries EOGH (horizontal) and EOGV (vertical) are computed as bipolar signals for the horizontal and vertical EOG channels, respectively. Adapted from [72] CC BY 4.0).

Participants walked on five different terrains in a single trial. The five locomotion modes were: level ground walking (LW), stair descent (SD), stair ascent (SA), ramp descent (RD), and ramp ascent (RA) (see Figure 3.3 for an overview). Prior to each walking stage, participants were instructed to stand still with eyes open for one minute to collect resting state EEG and EMG data. During the walking phase, participants were instructed to walk at a comfortable speed. An example of the synchronized data is shown in Figure 3.2. For a full description of the setup, one can refer to the published paper [72].

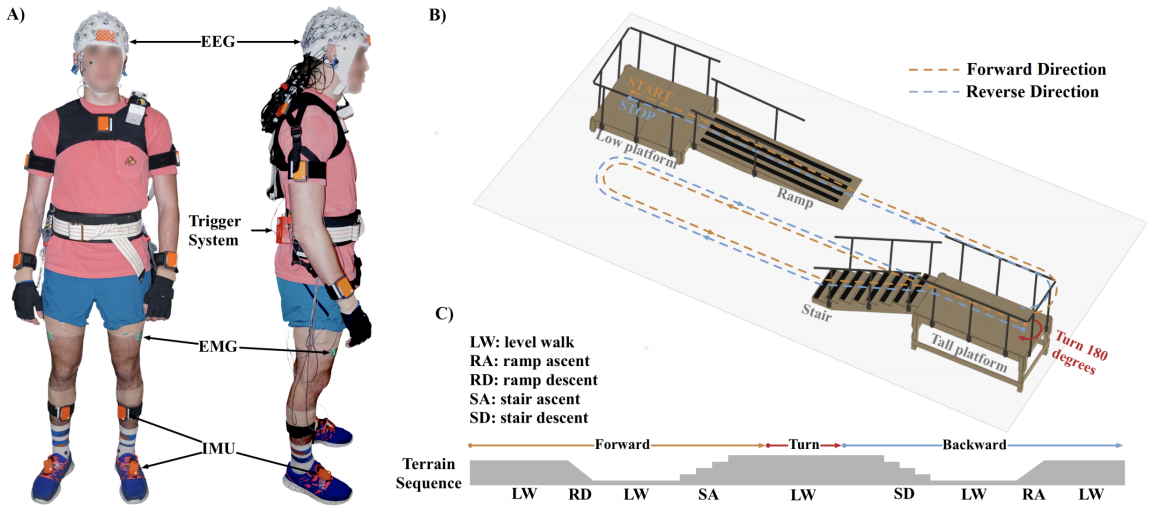

Figure 3.3: Experimental setup. A) Experimental setup. B) Illustration for gait course. C) Experimental protocol. Adapted from [21] CC BY 4.0).

3.3.2 Pre-processing and experimental designs

The data was first downsampled to 100 Hz after the removal of peripheral channels. Eye related artifacts were removed using the H-infinity algorithm with EOG channels as the noise reference [49]. ASR was used to remove bursts and motion noise from the data [73].

The EMG data were bandpass filtered using a 4th order butterworth filter from 30 to 300 Hz. The bandpass filtered signals were then rectified by taking the absolute value, then lowpass filtered at 6Hz using a 4th order butterworth filter to get the envelope of the EMG signals [74, 75].

Train, validation, test split

To generalize the decoder across subjects, a leave-one-out strategy was taken. There were two different experiments: 1) Intra- subject experiment and 2) Inter-

subject experiment. The first experiment aims to generalize the model within subjects across trials, generalizing towards unseen trials. The second aims to generalize the model across subjects. The illustration of each is illustrated in Figure 3.4. For example, the second inter- subject experiment left out one participant for the "test" set, two participants for "validation" and the rest of seven participants for "train". These combinations were rotated to train a model for each left out participant. In our offline experiments, the "train" data set was used to train the model with certain hyperparameters, the "validation" set was used to assess the hyperparameter combinations used in the "train" data set. Finally, the "test" data set was used to assess the best hyperparameter combinations determined by the "validation" set.

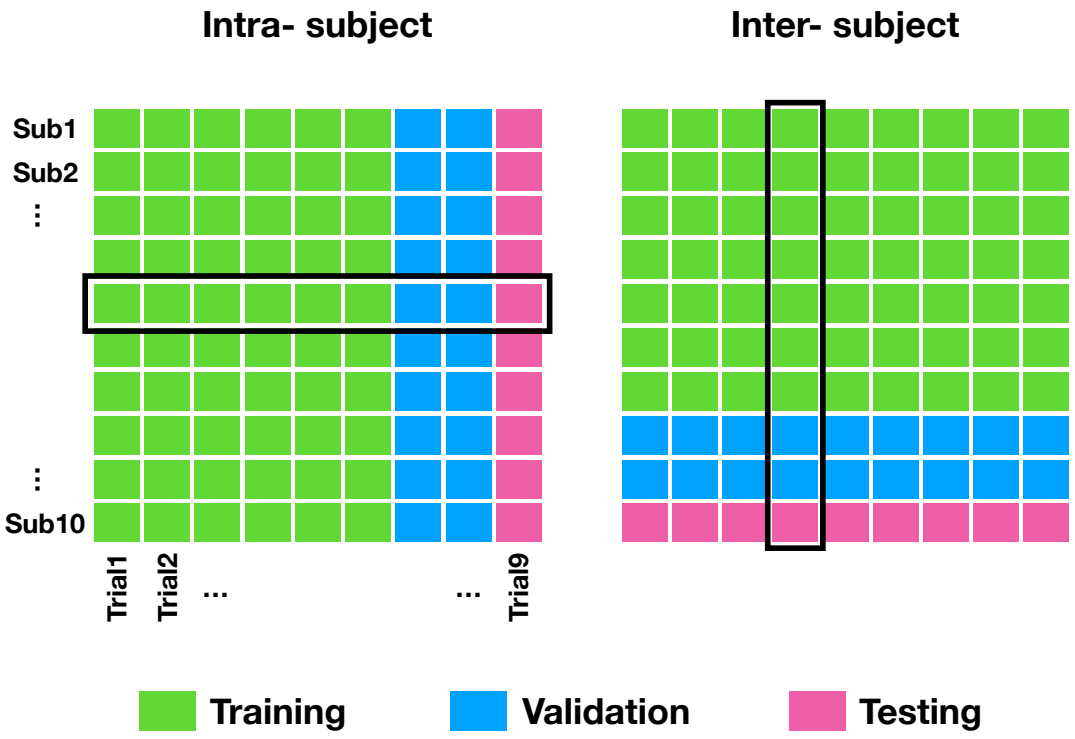

Figure 3.4: Illustration of intra- and inter- subject training schematic.

3.3.3 Code

The code is available on github: https://github.com/shonaka/EEG-generalized-decoding. To replicate the environment, the Anaconda virtual environment and the docker image are also available for replicating the building environment: `https://hub.docker.com/r/snakagome/research_gpu`.

3.3.4 Metrics

The same metrics were used to assess the performances as in the Aim 1 (Pearson's correlation coefficient = r-value and Coefficients of determination = R2 score).

3.3.5 Algorithms

Among the algorithms evaluated in specific aim 1, Linear Regression (LR), Ridge Regression (RR), Unscented Kalman Filter (UKF), Gated Recurrent Unit (GRU), Temporal Convolutional Network (TCN), and Quasi-Recurrent Neural Network (QRNN) were used to assess the performances. Some algorithms were removed due to similar characteristics, time to compute, and low performance that was not worth pursuing even further.

3.3.6 Hyperparameter optimization

The same hyperparameter optimization schema was used as in Aim 1 using the Optuna package. The number of trials was increased to 200 to incorporate more complex situations in the experimental design (overground walking with different gait speeds, five different complex terrains, etc).

3.3.7 Hybrid model

In a recent study, the authors showed a simple two-step approach to increase the accuracy and deal with subject variability in invasive speech decoding [71]. Inspired by this study, we also attempted to break down the decoding into two steps: 1) EEG to EMG and 2) EMG to kinematics, rather than direct EEG to kinematics, and compared the performance against each other. The illustration of this approach is illustrated in Figure 3.5.

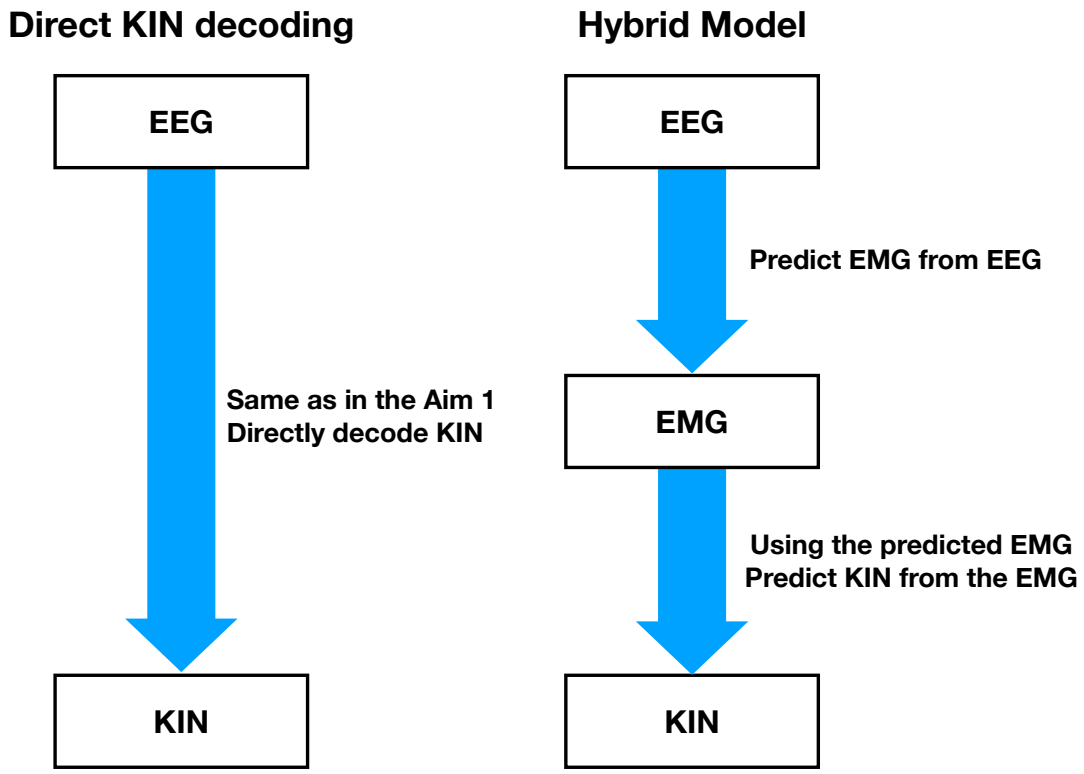

Figure 3.5: Illustration of a hybrid KIN decoding model inspired by the recent study on speech decoding [71].

3.3.8 Transfer learning

Two transfer learning on a inter- subject decoding model (EEG to KIN) was performed. In the most simple transfer learning, the inter- subject model was loaded with parameters frozen except for the final fully connected layer. This ensured the features learned throughout the training with different subjects' EEG data were retained. One trial from a subject is left out for testing. Another trial from the subject is selected to fine-tune the model. In the other transfer learning, the inter- subject model was loaded with all the parameters that were unfrozen to be retrained. In this case, the parameters learned through the inter- subject training was used to initialize the parameters. In both cases during fine-tuning, the total number of epochs was set to 10 with Adam optimizer with the learning rate of 1e-3 and other parameters set to default. Finally, the fine-tuned model was tested using the reserved trial to assess the performance of the transfer learning.

3.3.9 Statistical testing

To test the significance in difference for the effect of transfer learning, the Kruskal-Wallis test was used to assess the pairwise difference in distributions. The statistical test was chosen based on the fact that the underlying distribution of the metrics (r-value and R2 score) was not Gaussian.

3.4 Results

First, individual results of the decoding of intra- subjects are shown in section 3.4.1. Next, the inter- subjects performance across five different terrains is detailed in the section 3.4.2. To rule out the effect of complex experimental paradigms, we only extracted the level ground walking (LW) and retested the inter- subject decoding.

These results are shown in section 3.4.3. Further, to rule out the effect of different gait speeds in the inter- subject decoding, we tested the inter- subject decoding on the treadmill data (section 3.4.5). In another attempt to generalize the model, the hybrid two-step model was also tested to deal with inter- subject decoding in section 3.4.4. Lastly, the transfer learning results on the inter- subject for all terrains are shown in section 3.4.6. The summary of all the results is illustrated in Figure (Figure 3.20) in the discussion section.

3.4.1 Intra-subject Analysis

Intra-subject decoding analysis achieved good performance in decoding accuracy when predicting the unseen trial within the same subject. This is the most simple extension from Chapter 2 where EEG was used to predict the kinematics (the joint angles of the legs) while walking on a treadmill. The differences between Chapter 2 and the current results are that 1) the current results are based on intra- subject decoding where the decoder is generalized enough to deal with unseen trials. 2) the current dataset is not only overground walking, but also includes five different terrains, so the gait pattern changes across time.

EEG to KIN decoding

Intra- subject EEG to KIN decoding showed similar performance as compared to the previous results in Aim 1 and other published studies [21, 44] where a median r-value of 0.4 was observed (Figure 3.6). Focusing on the first row, which is the performance evaluated with r-value, a slight increase in performance was observed across different algorithms based on the increase in the tap sizes. The best performing algorithms were GRU and QRNN for hip, LR and QRNN for knee, and GRU for the ankle. We also observed similar low performance in the UKF when evaluated from R2 score, which is represented in the second row of Figure 3.6. TCN constantly

performed poorly when compared to other algorithms.

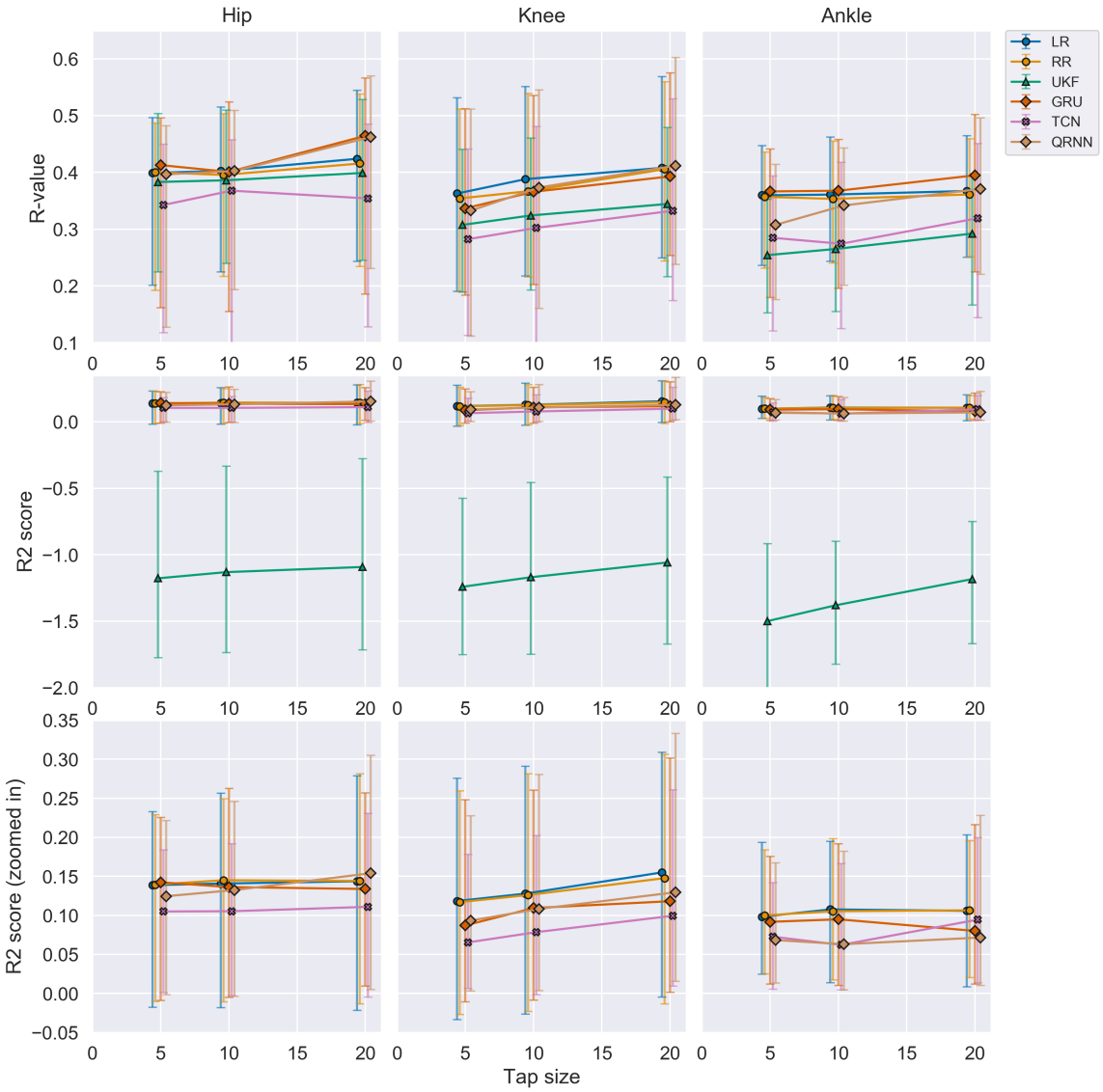

Figure 3.6: Overground Intra- subject EEG to KIN decoding. The third row is the same as the second row, but magnified to exclude UKF. The middle symbol indicates the median value, the lower bound shows the 25th percentile and the upper bound shows the 75th percentile.

EEG to EMG decoding

Intra- subject EEG to EMG showed lower performance compared to Intra- subject EEG to KIN. The performance of EEG to EMG was similar to the previous study

[74] where the median r-value range of 0.2 - 0.3 was observed (Figure 3.7). Again, we see a similar trend in performance where LR, RR, GRU, and QRNN perform better and UKF and TCN suffer from low performance. Slight increases in performance were observed with the increase in tap sizes in r-value. However, the performance dropped as we increased the tap sizes in linear decoders when evaluated from the R2 score perspective (third row in Figure 3.7).

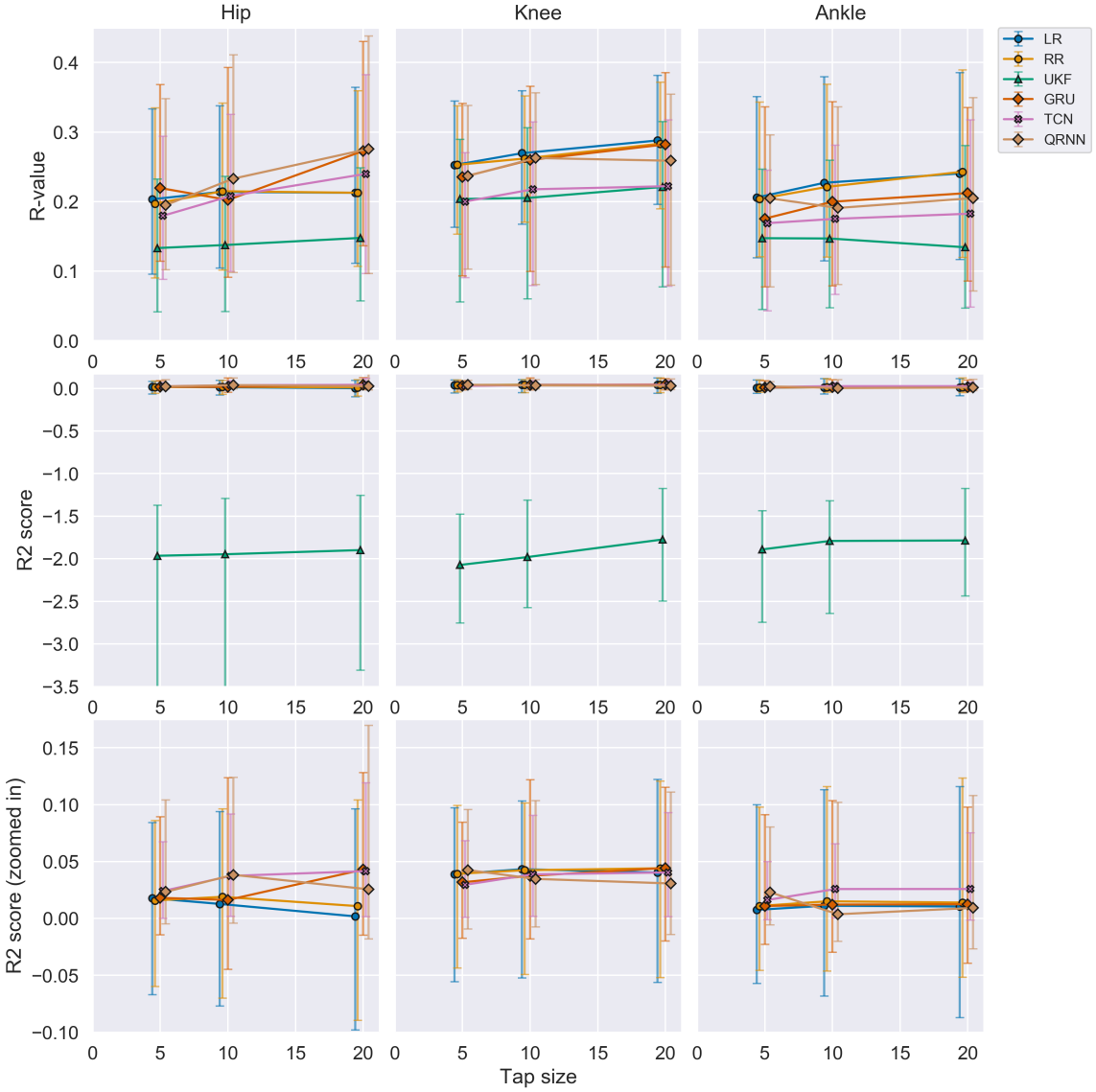

Figure 3.7: Overground Intra-subject EEG to EMG decoding. Refer to the Figure 3.6 for the explanations.

EMG to KIN decoding

Intra- subject EMG to KIN decoding showed the best performance in the intra-subject analysis (Figure 3.8). The performance was similar to our previous study [75] where the median r-value range of 0.7 - 0.8 was observed for the hip joint. Unlike the other intra- subject decoding results, intra- EMG to KIN decoding showed deep learning based algorithms (GRU, TCN, and QRNN) outperforming other algorithms, especially in the knee and ankle joints. The low performance in R2 score for the UKF was also not as significant as in the other intra- subject decoding results and the R2 scores were higher than -1. A slight increase in the decoding performance based on the tap sizes increase was also observed in this experiment.

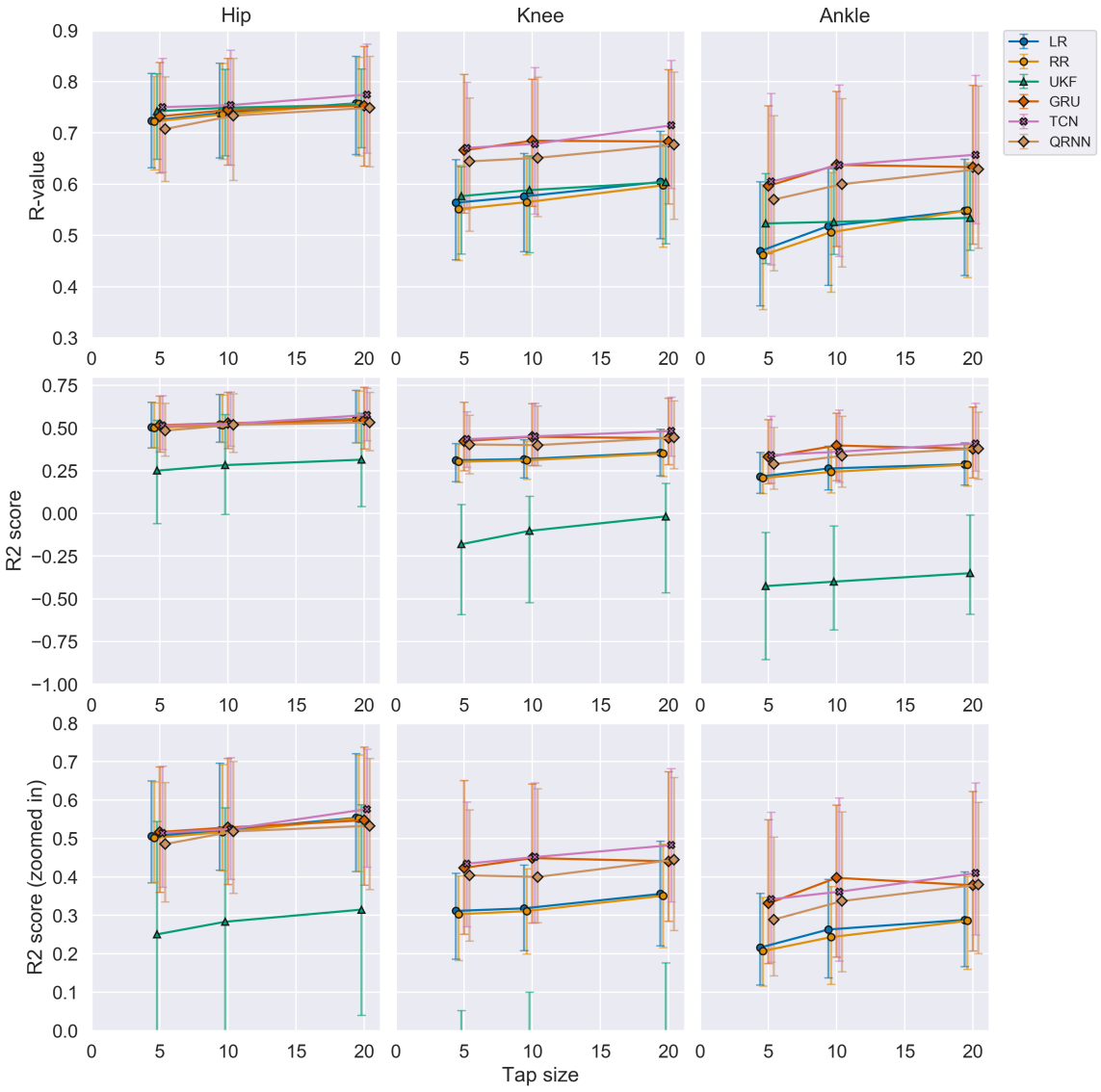

Figure 3.8: Overground Intra-subject EMG to KIN decoding. Refer to the Figure 3.6 for the explanations.

3.4.2 Inter-subject Analysis

Inter-subject decoding analysis showed low performance in the decoding accuracy predicting on the unseen subject. This is the most challenging decoding scheme as the model has to deal with subject variability in neural signals and complex experimental settings (different gait speed and five different terrains that require modification in

their gait patterns).

EEG to KIN decoding

Inter-subject EEG to KIN decoding suffered from low performance across all the algorithms and all the metrics (Figure 3.9). An interesting trend was a decrease in performance with respect to the increase in tap sizes. This was the opposite of what we observed in the intra-subject decoding results in general. When evaluated from R2 score, linear decoders specifically suffered a significant decrease in performance as the tap sizes increased (from -0.14 to -0.22 in median R2 scores). The deep learning decoders did not experience a decrease in performance for the current setup.

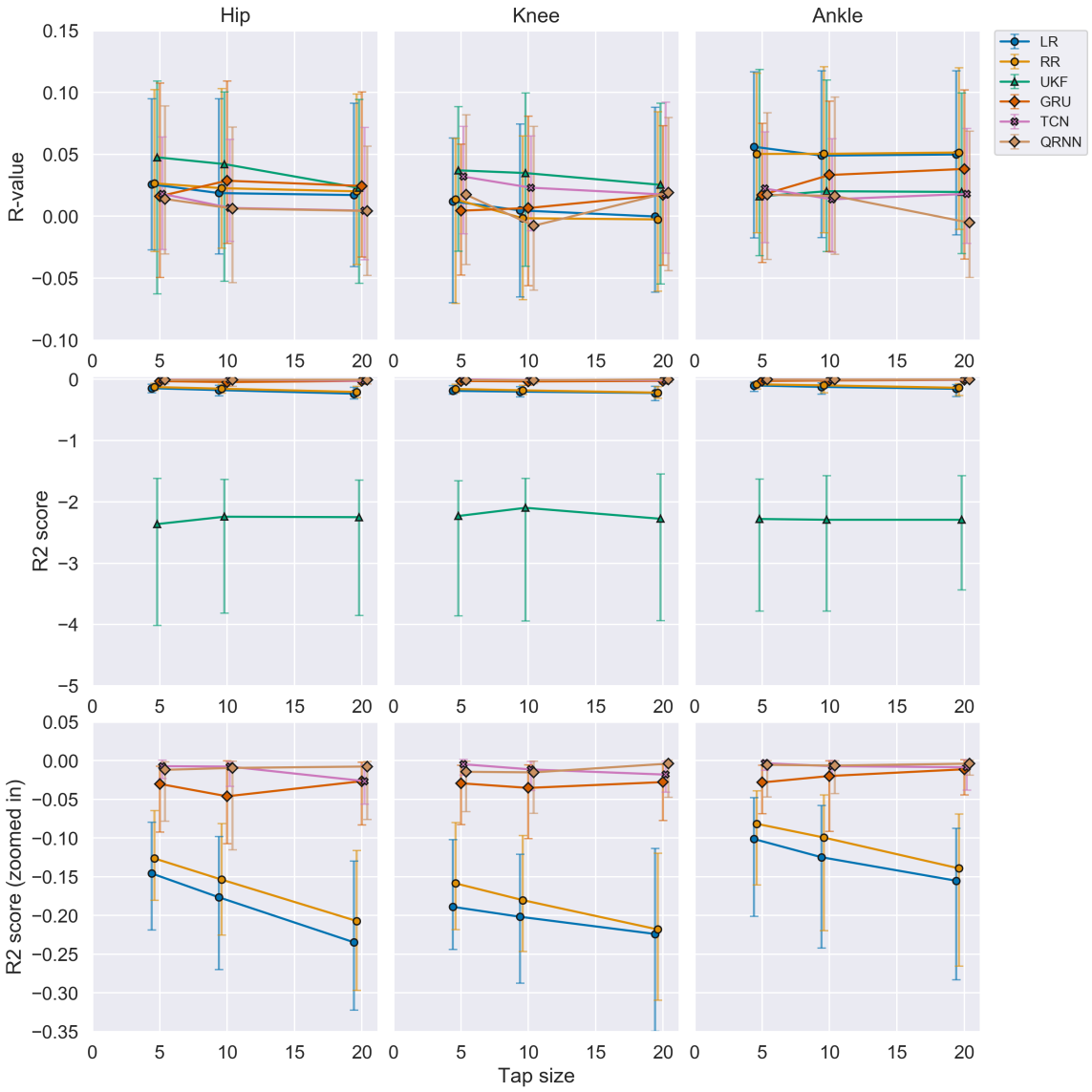

Figure 3.9: Overground Inter- subject EEG to KIN decoding. Refer to the Figure 3.6 for the explanations.

EEG to EMG decoding

Inter-subject EEG to EMG decoding also suffered from low performance (Figure 3.10) and showed a very similar trend in performance compared to the inter-subject EEG to KIN decoding. The trends were 1) low performance, 2) decrease in performance as the tap size increase, and 3) while linear decoders suffer from decreasing

performance, deep learning decoders were more robust.

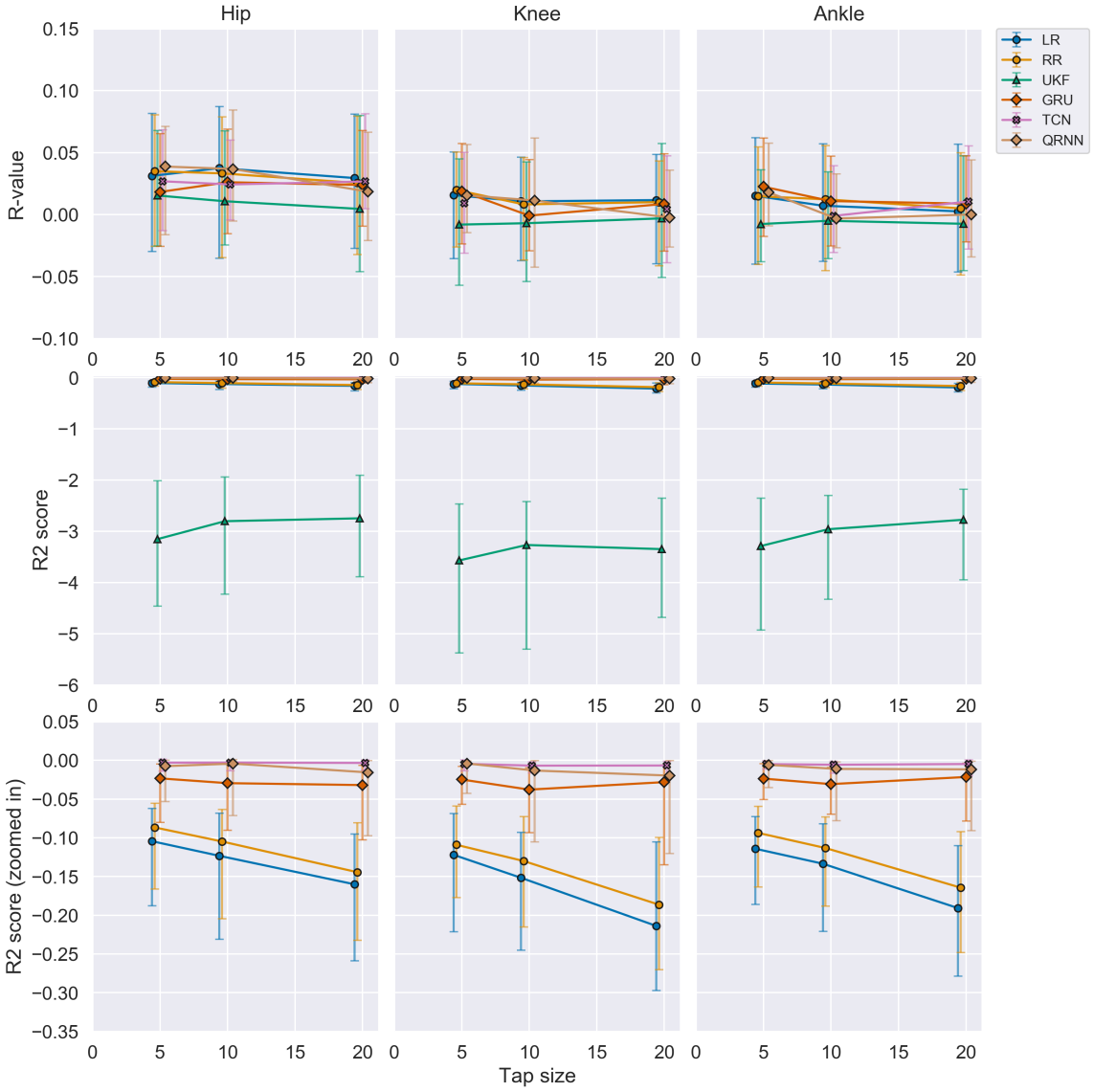

Figure 3.10: Overground Inter- subject EEG to EMG decoding

EMG to KIN decoding

The EMG to KIN (kinematics) decoding model showed a feasible performance. Despite the generalization across subjects (leave one subject out approach), the r-values showed median values of around 0.6, 0.4, and 0.3 for the hip, knee, and ankle, respectively. The R2 score showed median values of around 0.3, 0.2, and 0.05 for

hip, knee, and ankle, respectively. Figure 3.11 shows the whole picture of the results with median values and 25th to 75th percentile error bars. In this study, we clearly observed the performance difference among the joints where the hip joint decoding results were the best among all joints, whereas the ankle joint performed the worst. We also observed a similar R2 score decrease in the previous specific aim and in the published article [10].

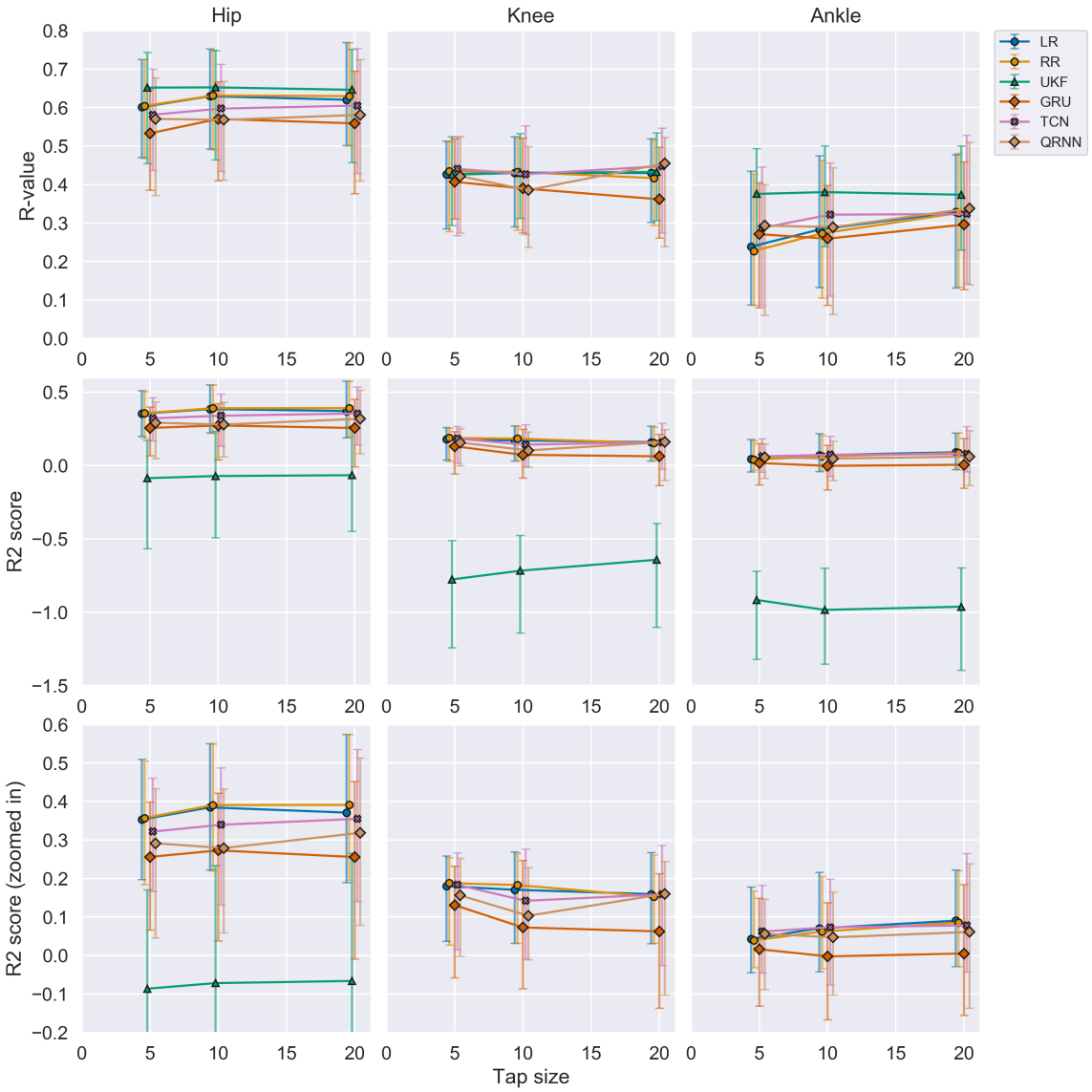

Figure 3.11: Overground Inter- subject EMG to KIN decoding. The third row is a zoomed in version of the second row with the same metric for better visualization. Each color corresponds to different algorithms.

3.4.3 Inter- subject Analysis (LW only)

It was difficult to rule out whether the low decoding accuracy in the inter- subject generalized model is due to the complexity in the experimental design (of having five different terrains in a self-paced overground walking situation) or the EEG subject variability. To rule out the effect of the complexity in the experimental design, only the level ground walking (LW) was extracted and the inter- subject analysis was reperformed in the same manner.

EEG to KIN decoding (LW only)

Even with the LW only inter- subject EEG to KIN decoding (Figure 3.12), we observed very similar low performance as compared to the inter- subject all five terrains decoding (Figure 3.9). Although there was a slight increase of around 0.05 in performance in the r-value compared to the all terrains included experiment, the performance was still low. The same performance decrease due to the tap size increase was observed in this result.

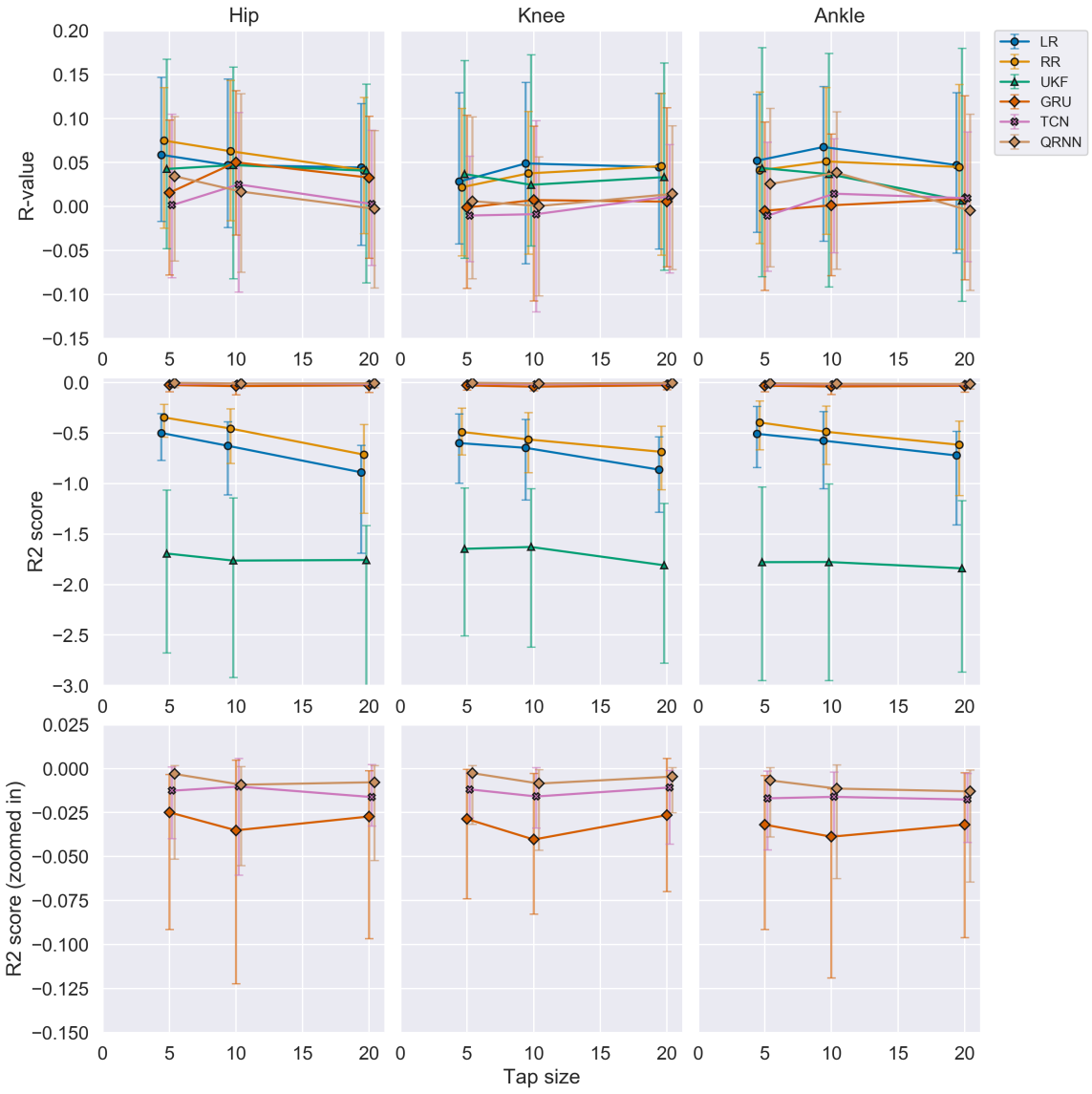

Figure 3.12: Overground Inter- subject EEG to KIN decoding (LW only)

EEG to EMG decoding (LW only)

Inter- subject decoding of EEG to EMG in LW only (Figure 3.13) also showed very similar low performance compared to the inter- subject all five terrains decoding (Figure 3.10).

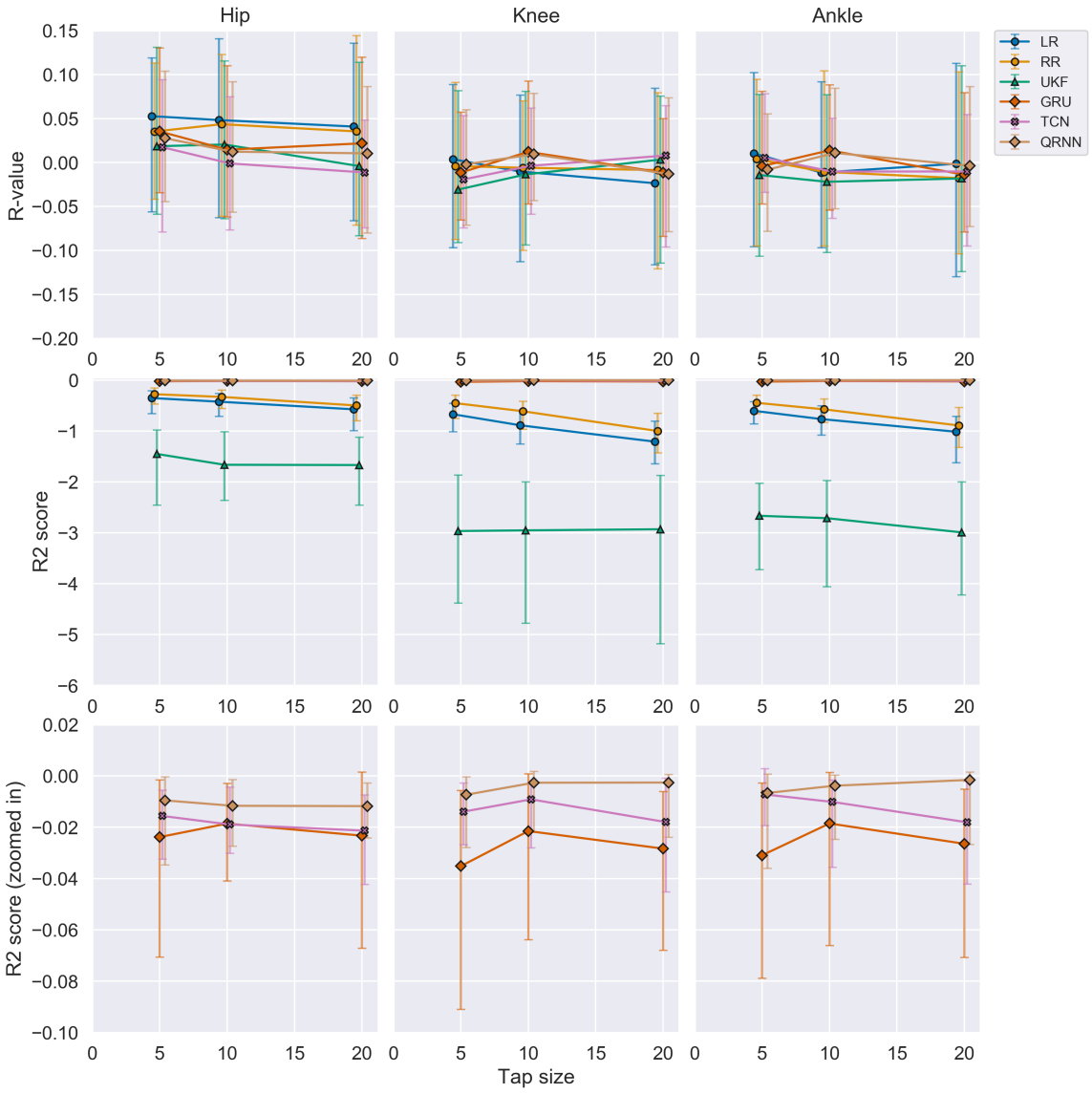

Figure 3.13: Overground Inter- subject EEG to EMG decoding (LW only)

EMG to KIN decoding (LW only)

Similar to inter-subject EMG to KIN decoding across all the terrains, the EMG to KIN decoding (LW only) showed good performance generalizing to a new unseen subject (Figure 3.14).

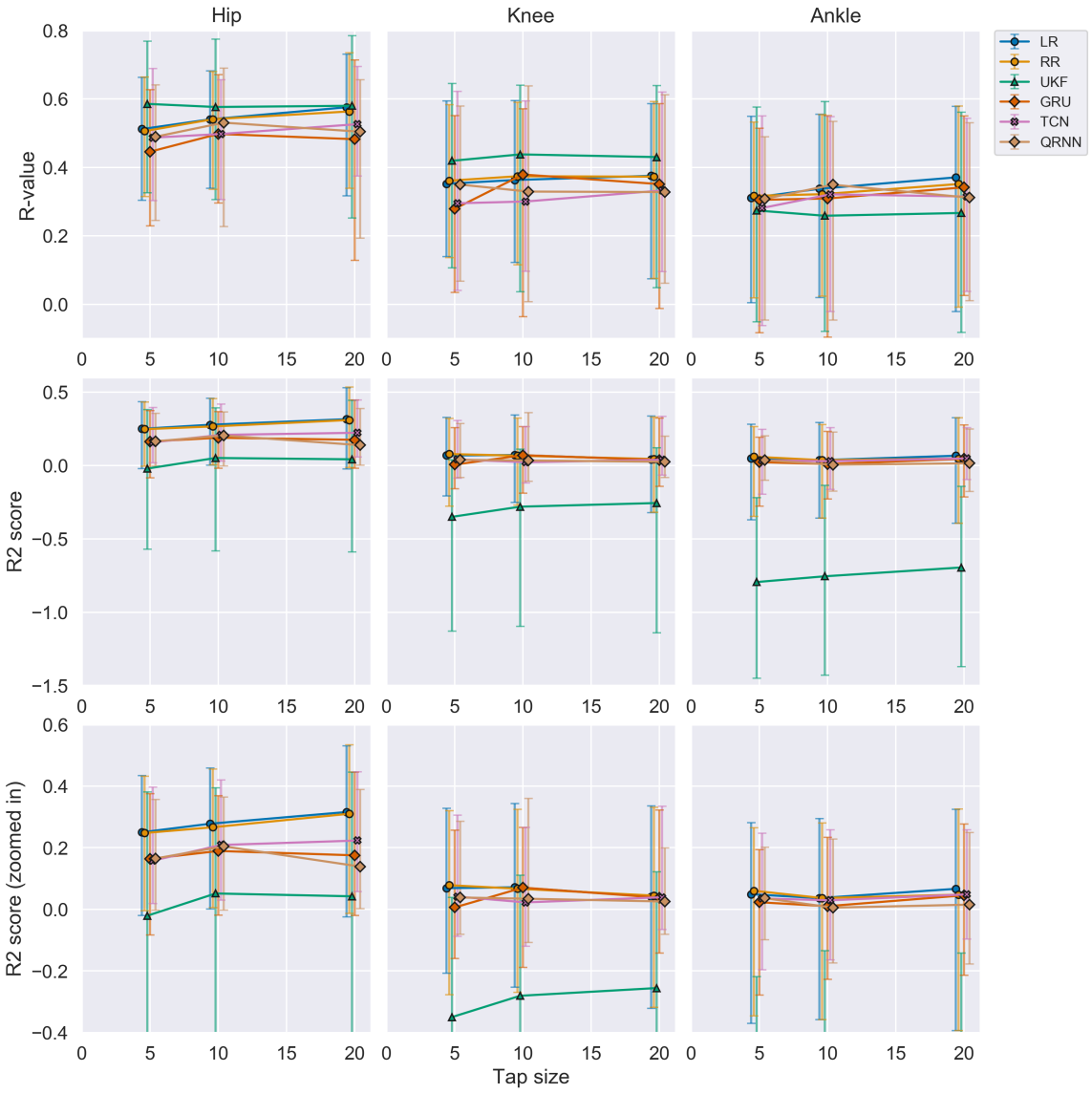

Figure 3.14: Overground Inter- subject EMG to KIN decoding (LW only)

3.4.4 Hybrid model analysis

Intra- 1) EEG to KIN vs 2) EEG to EMG to KIN

The intra- subject EEG to KIN performance was compared against a combination hybrid model where the EMG prediction from the intra- subject EEG to EMG was fed into the intra- subject EMG to KIN model to predict the KIN from the EEG.

The hybrid intra- subject EEG to KIN decoding showed relatively good performance in r-value, where three algorithms performed better than the median r-value of 0.3 in hip joint, and two algorithms around 0.2. The linear regression algorithm did not show feasible performance in any of the joints when evaluated from r-value perspective. When evaluated from R2 score, the performance across all the algorithms was relatively low compared to the intra- subject decoding of EMG to KIN (Figure 3.8).

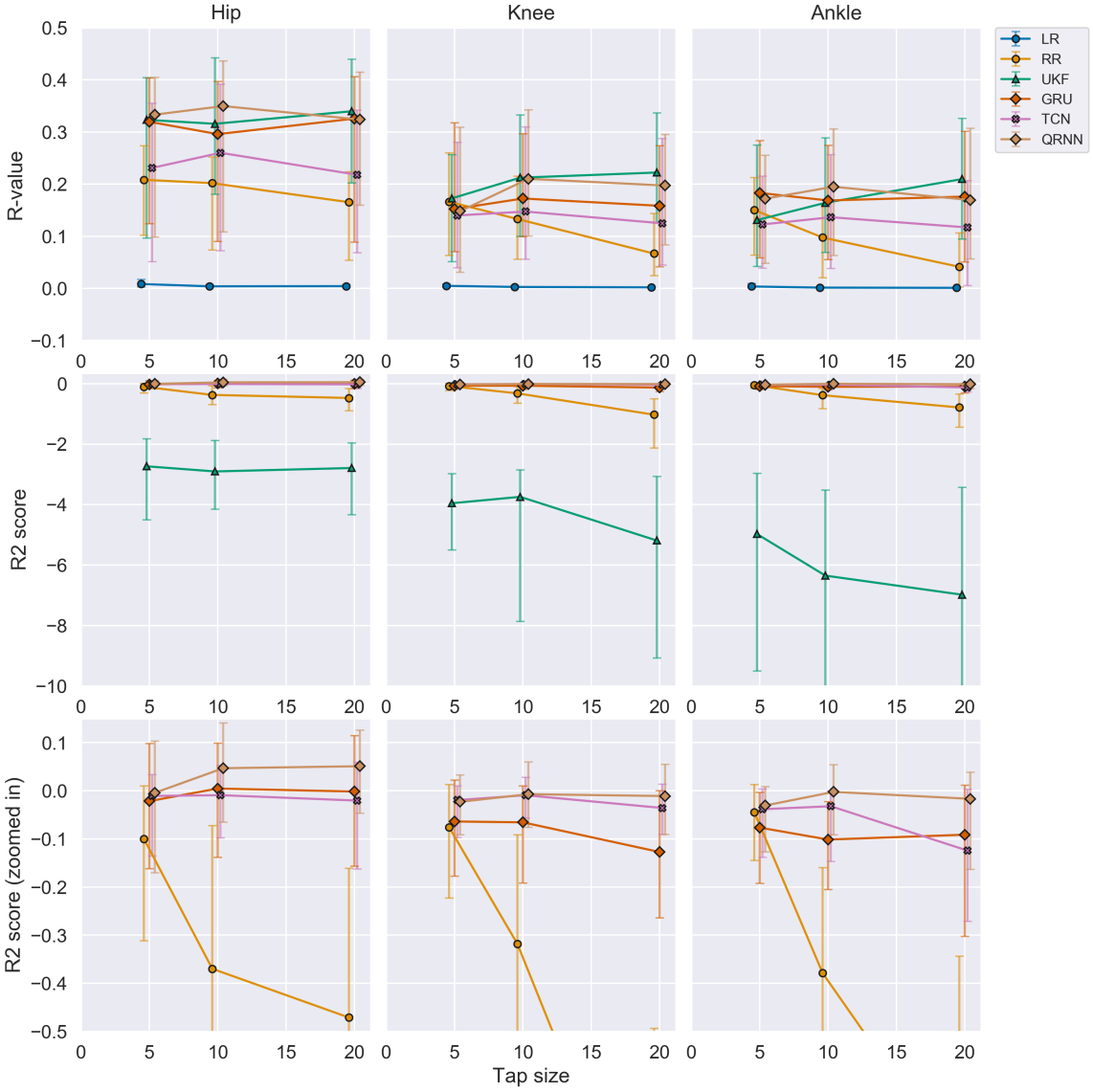

Figure 3.15: Overground Hybrid Intra- subject decoding

Inter- 1) EEG to KIN vs 2) EEG to EMG to KIN

The inter- subject EEG to KIN performance was compared against a combination hybrid model where the EMG prediction from the inter- subject EEG to EMG was fed into the inter- subject EMG to KIN model to predict the KIN from the EEG (Figure 3.16).

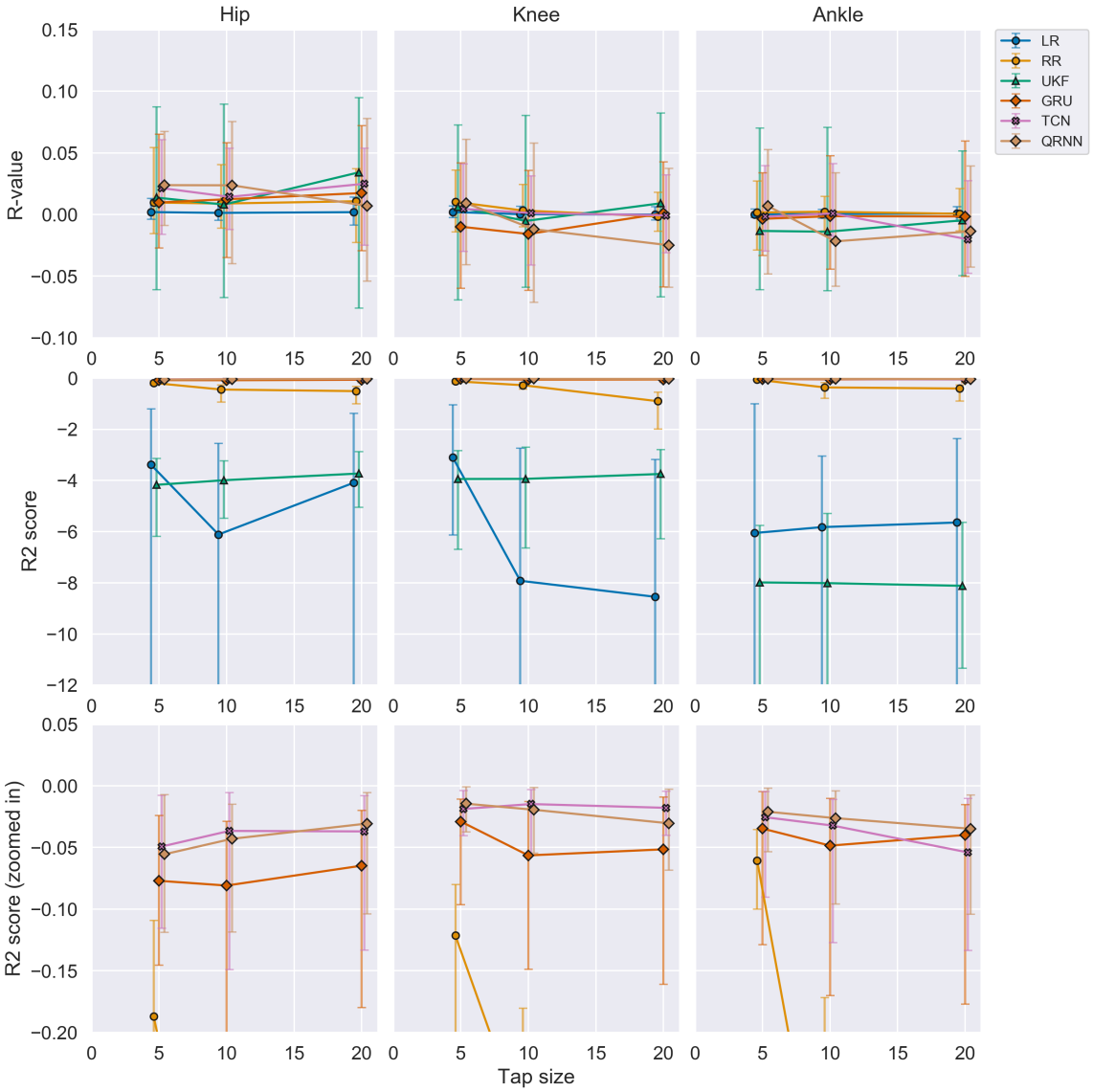

Figure 3.16: Overground Hybrid Inter- subject decoding.

3.4.5 Treadmill Inter- subject Analysis

To rule out the effect of the gait speed, we also conducted the same inter- subject decoding on a treadmill data from Aim 1. The performance showed similar low performance compared to all the other inter- subject decoding in the overground data (Figure 3.17).

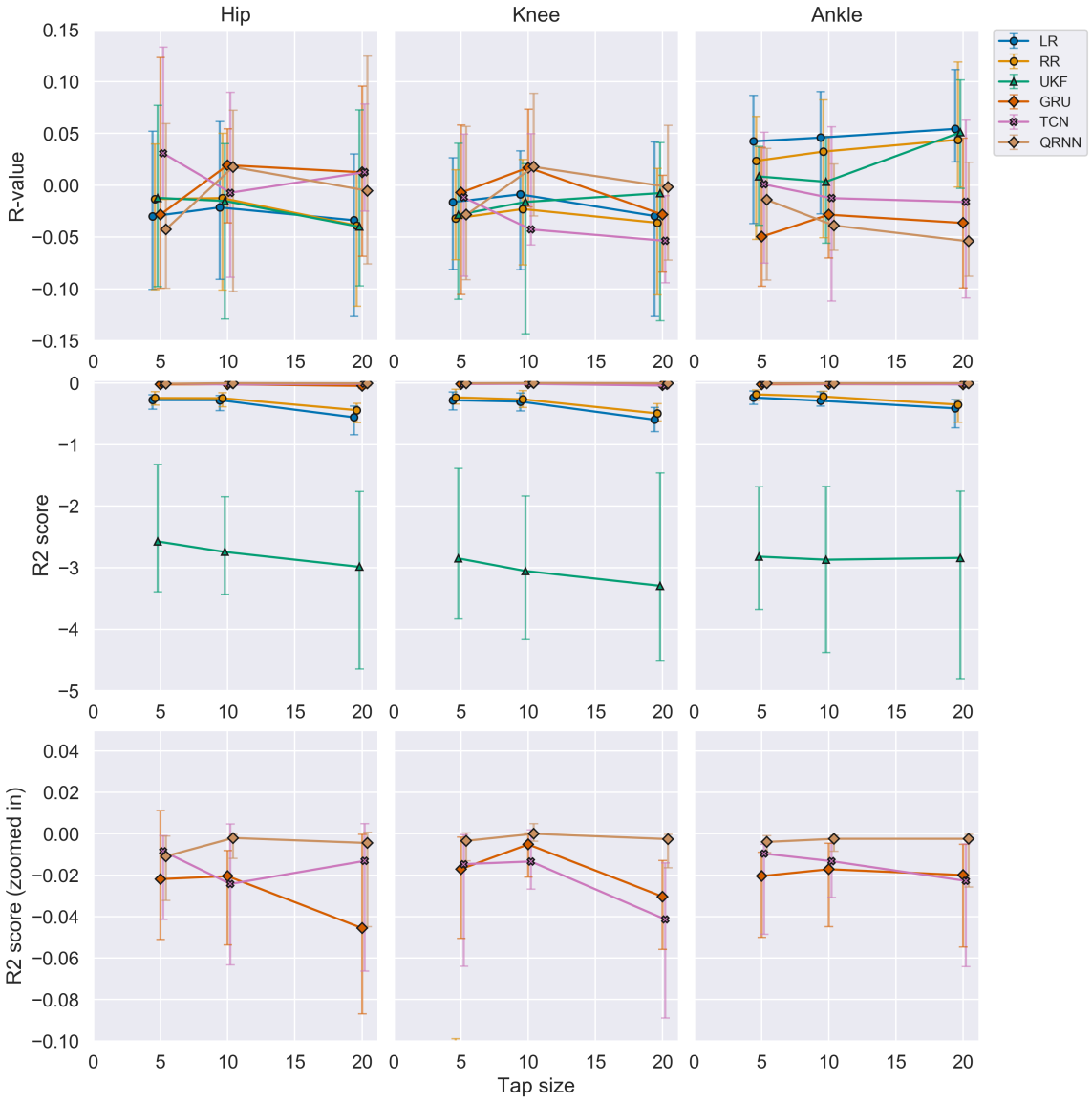

Figure 3.17: Treadmill Inter- subject decoding

3.4.6 Transfer learning

Transfer learning applied to inter- subject decoding on overground with all the terrains (EEG to KIN) significantly increased the performance in both R-value (Figure 3.18) and R2 score (Figure 3.19). Kruskal-Wallis test was used and all the pairwise comparison showed a significance in difference ($p < 0.05$). The R-values in the intra-subject decoding on EEG to KIN prediction had a median r-value of around 0.4 across all joints. The median r-values went down to near zero for inter- subject decoding across all the joints. Transfer learning applied to this inter- subject decoding significantly improved the decoding performance by increasing the median r-values to 1.5 - 1.8 across all the joints. (Figure 3.18).

The contributions of generalized features in transfer learning can be calculated by finding the ratio between the individual features contribution and generalized features contribution. Individual features Hip = 32.0%, Knee = 46.1%, Ankle = 38.3% (From Intra-subject comparison) and Hip = 42.1%, Knee = 49.4%, Ankle = 46.99% (From Transfer all comparison).

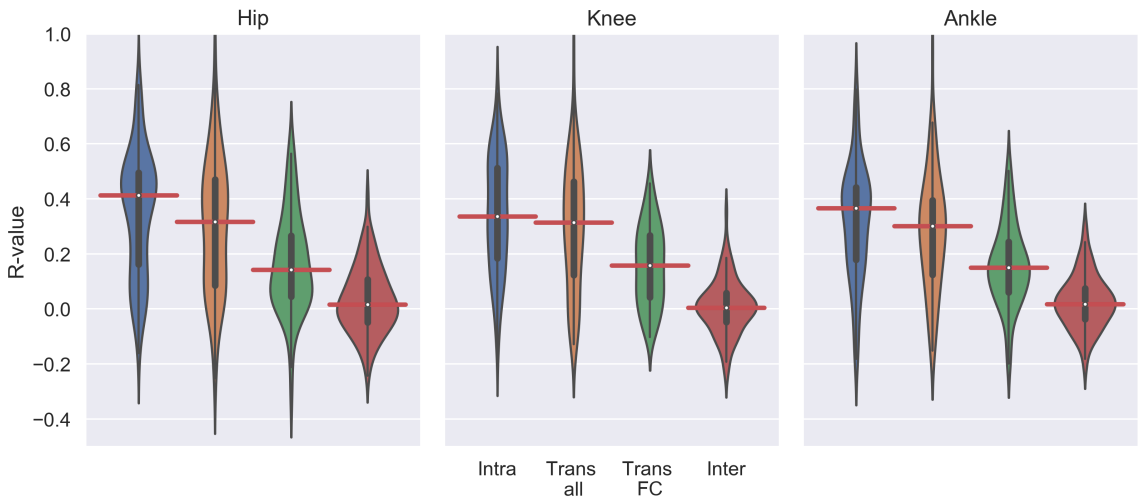

Figure 3.18: Overground transfer learning GRU tap size = 5 (R-value). The thin black line shows 10th to 90th percentile. The thick black line shows 25th to 75th percentile. The white dot and the red horizontal line also shows the median value and was added for easier comparisons with others.

The R2 scores in the intra- subject decoding on EEG to KIN prediction had a median r-value of around 0.1 - 0.15 across all the joints. The median r-values went down below zero for inter- subject decoding across all the joints. Transfer learning applied to this inter- subject decoding significantly improved the decoding performance by increasing the median r-values from near 0 to 0.2 across all the joints. Another notable point was that the transfer learning significantly decreased the variance of the performances across all the subjects (Figure 3.18). The contribution of transfer learning on the last fully connected layer (FC) with respect to the Intra-subject decoding were Hip = 27.4%, Knee = 41.1%, Ankle = 38.5%, respectively. The R2 score of transfer learning on all the layers was high in variance and low in median values of below zero.

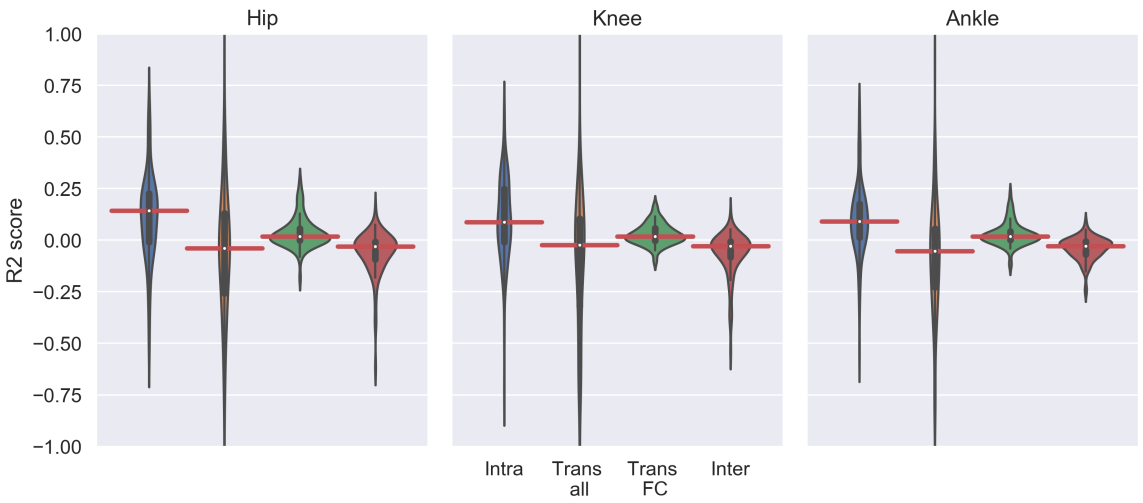

Figure 3.19: Overground transfer learning GRU tap size = 5 (R2 score).

Intra-subject decoding showed a feasible decoding performance. However, all the other inter-subject decoding performances showed low performances except for the EMG to KIN decoding. The only method to improve the inter-subject decoding performance was to use transfer learning. Since the transfer learning utilized generalized features learned during the inter-subject decoding, the performance improvements in the transfer learning can be regarded as contributions from the generalized features. On the other hand, the performance difference between the intra-subject decoding and the transfer learning can be regarded as individual features contributions. The summary of the generalization results is illustrated in Figure 3.20.

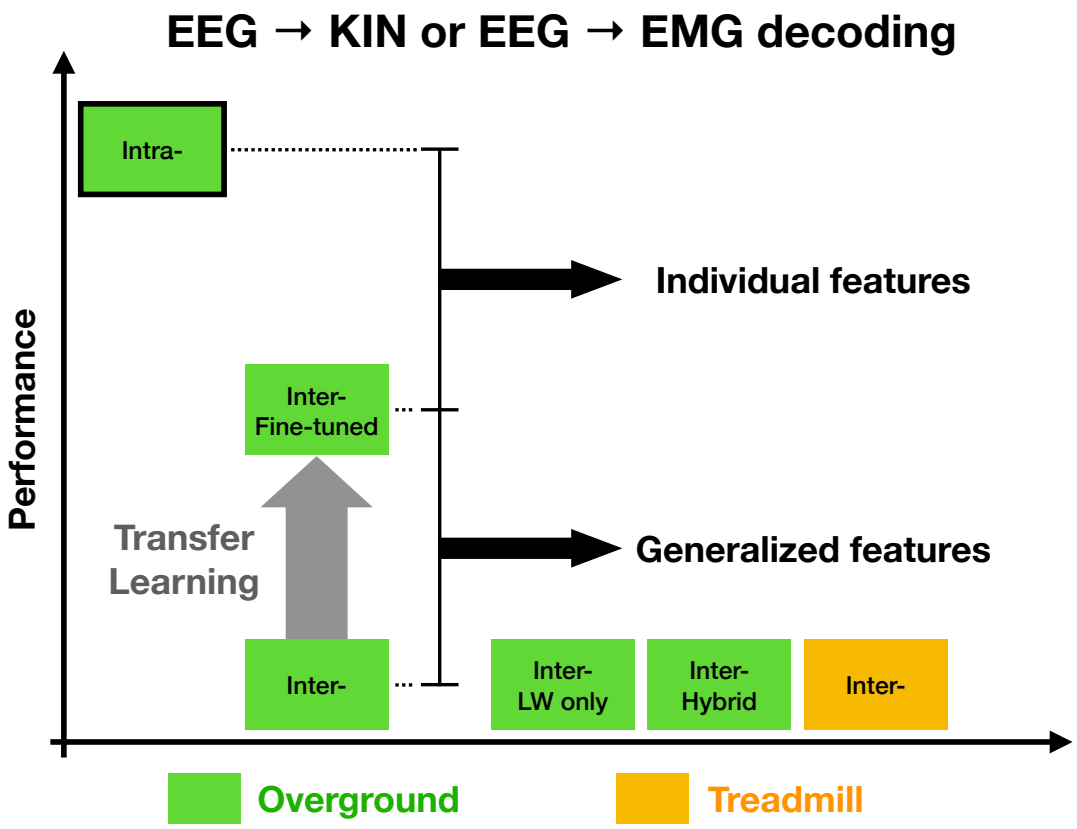

Figure 3.20: Summary of results comparing the performances (transfer FC). "Intra-" refers to section 3.4.1. "Inter- Fine-tuned" refers to section 3.4.6. "Inter-" refers to section 3.4.2. "Inter- LW only" refers to section 3.4.3. "Inter-Hybrid" refers to section 3.4.4. "Inter-" refers to section 3.4.5.

3.5 Discussion

In this study, we studied the ability to generalize lower limb gait decoding models within and across subjects. In the intra- subject decoding, where the generalization is across trials, all the combinations—1) EEG to KIN, 2) EEG to EMG, 3) EMG to KIN–showed good performances in the decoding. In the inter-subject decoding, where the generalization is across subjects, the performances were relatively low compared to that of the intra- subject decoding, except for EMG to KIN decoding. To improve the performance, a simpler experimental setup, treadmill walking for fixed gait speed,

and a hybrid two-step decoding were also assessed. However, these experiments did not improve the performance. Transfer learning, however, did significantly improve the performance of the inter- subject model, implying the features learned in the inter- subject model is still valuable for individual gait decoding despite the subject variability in EEG and gait speed.

Intra-subject decoding showed good performance across all the combinations: 1) EEG to KIN (Figure 3.6), 2) EEG to EMG (Figure 3.7), and 3) EMG to KIN (Figure 3.8). These results are a natural extension to the previous studies where EEG gait decoding is trained on the subject and trained on the same trial [10, 21, 44, 74, 75]. The current results showed it was possible to generalize the decoding models across trials even for simple linear decoders such as LR and RR. However, we also saw the low performance in the R2 score for UKF as we have shown in the previous study [10]. This indicates that UKF should be used with caution if the decoding requires more precise decoding with the minimal mean squared error between the predicted values and the actual values.

Inter-subject decoding showed low performance in the combinations: 1) EEG to KIN and 2) EEG to EMG, but showed a good performance in 3) EMG to KIN decoding. This implied that the decoding involving EEG was challenging in the inter-subject decoding context. However, EMG to KIN decoding could be used as a generalized decoding model despite the most complex overground settings with five different terrains.

To understand the origin of the low performance in the inter-subject decoding, we first eliminated the complexity in the experimental design by only focusing on the level ground walking (section 3.4.3). Despite the change, the inter-subject decoding performance with EEG barely improved its performance.

As an alternative approach to generalize the model, a recent study on speech decoding employed a hybrid two-step decoding [71]. Incorporating this approach,

we also employed the two step decoding starting from EEG to EMG and using the predicted EMG to decode KIN (section 3.4.4). However, this approach in this lower limb decoding also did not improve the performance compared to the inter-subject decoding in section 3.4.2.

Furthermore, to rule out not only the complexity in the walking settings, but to also fix the walking speed, we applied the same inter-subject decoding approach to the treadmill walking data utilized in the Aim 1. Despite being one of the most simple setups for gait decoding, this also did not improve the performance of inter-subject decoding.

These results imply that the individual variability of features in EEG is larger than expected, making the inter-subject decoding more challenging. This subject variability in EEG has been known for a long time, but little is known about the characteristics or origins of its variability [76]. In fact, subject variability in EEG is large enough to identify individuals from a sensor level EEG signal features [77]. It is worth mentioning that one needs to be cautious when the inter-subject decoding performance in EEG was high in performance because that may indicate the preprocessing in EEG is not adequate and the movement noise could be the dominant factor for the high decoding accuracy.

In terms of simpler linear decoders such as LR and RR performing as better as the neural networks model, a recent study reported a similar pattern in a time series benchmarking performance where statistical models outperforming neural networks [78]. The study used a relatively small dataset and the performance may be due to the overfitting, but this could similar in BCI studies. In fact, GRU or QRNN performing better than the LSTM in aim 1 implies a simpler model would be suitable in our experiments.

In the EEG decoding classification tasks, one of the most promising approaches to deal with the subject variability is to use transfer learning (See a recent review

on transfer learning on EEG-based BCI [79]). The transfer learning results in this paper showed promising results, increasing the inter-subject decoding performance significantly (section 3.4.6). As shown in Figure 3.20, from transfer learning on the fully-connected layer, we could conclude that the improvement primarily came from the generalizable features learned in the inter-subject decoding model and the remaining performance increase compared to the intra-subject decoding model could be the individual subject features in EEG. This is because the only layer that was fine-tuned with the transfer learning is the last fully-connected layer and the intermediate recurrent or convolutional layers were fixed with parameters from the inter-subject decoding training. Furthermore, the transfer learning on all the layers showed a significant performance increase in r-value, but showed an increase in variance and low median R2 score. Some reasons for this may be due to the fact a fixed small number of epochs across different subjects were used unlike in the other experiments where early stopping with validation was used. The tradeoff between transfer learning computation speed and the performance should be investigated more in the future, but even a small number of epochs could still increase the performance in inter-subject decoding.

The current study showed an initial step towards generalizing the model across trials and subjects. There are still many ways the generalization could be studied as indicated in the EEG classification tasks studies [8]. One of the promising methods discussed in the paper was to use the Riemannian geometry which is part of a metric learning [80]. In addition, a recent study showed that adding an adversarial attack to a model to avoid overfitting to individual features could improve the generalizability of the model [81]. Although these are some of the promising methods, it is still not clear if we could use this approach in a continuous decoding context as regression-based tasks are generally more challenging compared to classification tasks.

Future studies should explore: 1) other generalizations, 2) include some data from

patients (e.g., stroke patients, spinal cord injury patients, etc.), and 3) collect both treadmill and overground with the same subjects. As mentioned in the above limitations, some other promising generalization methods should be explored. It is also of importance to investigate patients' data to see if there are any changes in performance in terms of generalization can be observed. At last, to further investigate the differences and commonality between the treadmill walking and overground walking, collecting datasets with the same subjects on these two modalities are important. Regarding this, an interesting topic to pursue is to use meta-learning [82] to learn the common representation between the treadmill walking and overground walking.

Chapter 4

Conclusion

The purpose of this dissertation was focused on understanding and building a foundation for better decoding algorithms for gait. In the first chapter, we introduced an overview of the current problems and past attempts towards improving the decoding from both performance and robustness perspective. As mentioned in the chapter, current issues raised the necessity of 1) benchmarking and modeling and 2) generalization in gait decoding. To address these issues, the second chapter showed rigorous comparisons of different experimental paradigms and algorithms along with parameters. The study showed neural networks based algorithms were better in performance and more robust to external disturbance. To address the generalization problem, the third chapter studied generalization from intra- to inter- subject perspectives. Although the generalization across trials for intra-subject was feasible in all the combinations, in the inter-subject decoding where the decoder has to generalize for an unseen subject, EEG subject variability was still the largest issue. We also showed how transfer learning could be the key to improve the decoding performance in this generalization scenario. The work presented in this thesis contributes to the foundation of gait decoding.

References

[1] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[2] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *Journal of neurophysiology*, vol. 94, no. 1, pp. 8–25, 2005.

[3] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. Nicolelis, "Learning to control a brain–machine interface for reaching and grasping by primates," *PLoS biology*, vol. 1, no. 2, 2003.

[4] W. Wu, M. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. Donoghue, "Inferring hand motion from multi-cell recordings in motor cortex using a kalman filter," in *SAB'02-workshop on motor control in humans and robots: On the interplay of real brains and artificial devices*, pp. 66–73, 2002.

[5] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a kalman filter," *Neural computation*, vol. 18, no. 1, pp. 80–118, 2006.

[6] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Levedev, C. S. Henriquez, and M. A. Nicolelis, "Unscented Kalman filter for brain-machine interfaces," *PLoS ONE*, vol. 4, no. 7, 2009.

[7] M. A. Lebedev and M. A. Nicolelis, "Brain-machine interfaces: From basic science to neuroprostheses and neurorehabilitation," *Physiological reviews*, vol. 97, no. 2, pp. 767–837, 2017.

[8] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for eeg-based brain–computer

interfaces: a 10 year update," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.

[9] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic review," *Journal of neural engineering*, vol. 16, no. 5, p. 051001, 2019.

[10] S. Nakagome, T. P. Luu, Y. He, A. S. Ravindran, and J. L. Contreras-Vidal, "An empirical comparison of neural networks and machine learning algorithms for eeg gait decoding," *Scientific Reports*, vol. 10, no. 1, pp. 1–17, 2020.

[11] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, "Brain-computer interface technology: A review of the first international meeting," *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 2, pp. 164–173, 2000.

[12] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain–computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[13] 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, pp. 63–71, 2004.

[14] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, p. R1, 2007.

[15] S. Makeig, K. Gramann, T.-P. Jung, T. J. Sejnowski, and H. Poizner, "Linking brain, mind and behavior," *International Journal of Psychophysiology*, vol. 73, no. 2, pp. 95–100, 2009.

[16] J. T. Gwin, K. Gramann, S. Makeig, and D. P. Ferris, "Removal of movement artifact from high-density EEG recorded during walking and running," *Journal of Neurophysiology*, vol. 103, no. 6, pp. 3526–3534, 2010.

[17] A. H. Do, P. T. Wang, C. E. King, S. N. Chun, and Z. Nenadic, "Brain-computer interface controlled robotic gait orthosis," *Journal of neuroengineering and rehabilitation*, vol. 10, no. 1, p. 111, 2013.

[18] A. Kilicarslan, S. Prasad, R. G. Grossman, and J. L. Contreras-Vidal, "High accuracy decoding of user intentions using EEG to control a lower-body exoskeleton," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pp. 5606–5609, 2013.

[19] M. Seeber, R. Scherer, J. Wagner, T. Solis-Escalante, and G. R. Müller-Putz, "High and low gamma EEG oscillations in central sensorimotor areas are conversely modulated during the human gait cycle," *NeuroImage*, vol. 112, pp. 318–326, 2015.

[20] J. Wagner, S. Makeig, M. Gola, C. Neuper, and G. Müller-Putz, "Distinct β band oscillatory networks subserving motor and cognitive control during gait adaptation," *Journal of Neuroscience*, vol. 36, no. 7, pp. 2212–2226, 2016.

[21] T. P. Luu, J. A. Brantley, S. Nakagome, F. Zhu, and J. L. Contreras-Vidal, "Electrocortical correlates of human level-ground, slope, and stair walking," *PLoS ONE*, vol. 12, no. 11, 2017.

[22] B. H. Dobkin, "Brain-computer interface technology as a tool to augment plasticity and outcomes for neurological rehabilitation," *Journal of Physiology*, vol. 579, no. 3, pp. 637–642, 2007.

[23] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-Inspired Artificial Intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[24] J. I. Glaser, A. S. Benjamin, R. Farhoodi, and K. P. Kording, "The roles of supervised machine learning in systems neuroscience," *Progress in Neurobiology*, vol. 175, pp. 126–137, 2019.

[25] O. Barak, "Recurrent neural networks as versatile tools of neuroscience research," *Current opinion in neurobiology*, vol. 46, pp. 1–6, 2017.

[26] G. Blohm, K. P. Kording, and P. R. Schrater, "A how-to-model guide for neuroscience," *eNeuro*, vol. 7, no. 1, 2020.

[27] M. Wessels, C. Lucas, I. Eriks-Hoogland, and S. De Groot, "Body weight-supported gait training for restoration of walking in people with an incomplete spinal cord injury: A systematic review," *Assistive Technology Research Series*, vol. 26, no. 6, pp. 297–299, 2010.

[28] G. Kwakkel, B. J. Kollen, and R. C. Wagenaar, "Therapy impact on functional recovery in stroke rehabilitation," *Physiotherapy*, vol. 85, no. 7, pp. 377–391, 1999.

[29] T. C. Bulea, A. Kilicarslan, R. Ozdemir, W. H. Paloski, and J. L. Contreras-Vidal, "Simultaneous scalp electroencephalography (EEG), electromyography (EMG), and whole-body segmental inertial recording for multi-modal neural decoding," *Journal of visualized experiments : JoVE*, no. 77, pp. 1–13, 2013.

[30] T. P. Luu, Y. He, S. Nakagome, K. Nathan, S. Brown, J. Gorges, and J. L. Contreras-Vidal, "Multi-trial gait adaptation of healthy individuals during visual kinematic perturbations," *Frontiers in Human Neuroscience*, vol. 11, 2017.

[31] J. I. Glaser, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding," *arxiv preprint arXiv:1708.00909*, 2017.

[32] A. S. Benjamin, H. L. Fernandes, T. Tomlinson, P. Ramkumar, C. VerSteeg, R. H. Chowdhury, L. E. Miller, and K. P. Kording, "Modern machine learning as a

benchmark for fitting neural responses," *Frontiers in computational neuroscience*, vol. 12, p. 56, 2018.

[33] A. I. Sburlea, L. Montesano, and J. Minguez, "Continuous detection of the self-initiated walking pre-movement state from EEG correlates without session-to-session recalibration," *Journal of Neural Engineering*, vol. 12, no. 3, 2015.

[34] F. Alton, L. Baldey, S. Caplan, and M. C. Morrissey, "A kinematic comparison of overground and treadmill walking," *Clinical Biomechanics*, vol. 13, no. 6, pp. 434–440, 1998.

[35] S. J. Lee and J. Hidler, "Biomechanics of overground vs. treadmill walking in healthy individuals," *Journal of applied physiology*, vol. 104, no. 3, pp. 747–755, 2008.

[36] S. L. Chiu, C. C. Chang, and L. S. Chou, "Inter-joint coordination of overground versus treadmill walking in young adults," *Gait and Posture*, vol. 41, no. 1, pp. 316–318, 2015.

[37] K. Reinecke, M. Cordes, C. Lerch, F. Koutsandréou, M. Schubert, M. Weiss, and J. Baumeister, "From lab to field conditions: A pilot study on EEG methodology in applied sports sciences," *Applied Psychophysiology Biofeedback*, vol. 36, no. 4, pp. 265–271, 2011.

[38] M. Volker, R. T. Schirrmeister, L. D. Fiederer, W. Burgard, and T. Ball, "Deep transfer learning for error decoding from non-invasive EEG," *2018 6th International Conference on Brain-Computer Interface, BCI 2018*, vol. 2018-Janua, pp. 1–6, 2018.

[39] M. Paluš, "Nonlinearity in normal human eeg: cycles, temporal asymmetry, non-stationarity and randomness, not chaos," *Biological cybernetics*, vol. 75, no. 5, pp. 389–396, 1996.

[40] J. M. Antelis, L. Montesano, A. Ramos-Murguialday, N. Birbaumer, and J. Minguez, "On the Usage of Linear Regression Models to Reconstruct Limb Kinematics from Low Frequency EEG Signals," *PLoS ONE*, vol. 8, no. 4, 2013.

[41] T. Castermans, M. Duvinage, G. Cheron, and T. Dutoit, "About the cortical origin of the low-delta and high-gamma rhythms observed in EEG signals during treadmill walking," *Neuroscience Letters*, vol. 561, pp. 166–170, 2014.

[42] A. Presacco, R. Goodman, L. Forrester, and J. L. Contreras-Vidal, "Neural decoding of treadmill walking from noninvasive electroencephalographic signals," *Journal of Neurophysiology*, vol. 106, no. 4, pp. 1875–1887, 2011.

[43] A. Presacco, L. W. Forrester, and J. L. Contreras-Vidal, "Decoding intra-limb and inter-limb kinematics during treadmill walking from scalp electroencephalographic (EEG) signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 2, pp. 212–219, 2012.

[44] T. P. Luu, S. Nakagome, Y. He, and J. L. Contreras-Vidal, "Real-time EEG-based brain-computer interface to a virtual avatar enhances cortical involvement in human treadmill walking," *Scientific Reports*, vol. 7, no. 1, 2017.

[45] T. P. Luu, Y. He, S. Brown, S. Nakagame, and J. L. Contreras-Vidal, "Gait adaptation to visual kinematic perturbations using a real-time closed-loop brain-computer interface to a virtual reality avatar," *Journal of Neural Engineering*, vol. 13, no. 3, p. 36006, 2016.

[46] T. P. Luu, Y. He, S. Brown, S. Nakagome, and J. L. Contreras-Vidal, "A closed-loop brain computer interface to a virtual reality avatar: Gait adaptation to visual kinematic perturbations," in *International Conference on Virtual Rehabilitation, ICVR*, pp. 30–37, 2015.

[47] Y. He, K. Nathan, A. Venkatakrishnan, R. Rovekamp, C. Beck, R. Ozdemir, G. E. Francisco, and J. L. Contreras-Vidal, "An integrated neuro-robotic interface

for stroke rehabilitation using the nasa x1 powered lower limb exoskeleton," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3985–3988, IEEE, 2014.

[48] H. Yokoyama, N. Kaneko, T. Ogawa, N. Kawashima, K. Watanabe, and K. Nakazawa, "Cortical correlates of locomotor muscle synergy activation in humans: An electroencephalographic decoding study," *iScience*, vol. 15, pp. 623–639, 2019.

[49] A. Kilicarslan, R. G. Grossman, and J. L. Contreras-Vidal, "A robust adaptive denoising framework for real-time artifact removal in scalp EEG measurements," *Journal of Neural Engineering*, vol. 13, no. 2, 2016.

[50] Y. He, T. P. Luu, K. Nathan, S. Nakagome, and J. L. Contreras-Vidal, "A mobile brain-body imaging dataset recorded during treadmill walking with a brain-computer interface," *Scientific data*, vol. 5, 2018.

[51] A. H. Fagg, G. W. Ojakangas, L. E. Miller, and N. G. Hatsopoulos, "Kinetic trajectory decoding using motor cortical ensembles," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 5, pp. 487–496, 2009.

[52] G. A. Seber and A. J. Lee, *Linear regression analysis*, vol. 329. John Wiley & Sons, 2012.

[53] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[54] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*, vol. 3068, pp. 182–193, International Society for Optics and Photonics, 1997.

[55] A. V. Dorogush, V. Ershov, and A. Gulin, "Catboost: gradient boosting with categorical features support," *arXiv preprint arXiv:1810.11363*, 2018.

[56] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, ACM, 2016.

[57] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, pp. 3146–3154, 2017.

[58] 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.

[59] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," in *Proceedings of the 7th International Conference on Learning Representations*, (New Orleans, Louisiana), May 2019.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[61] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[62] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International Conference on Machine Learning*, pp. 2342–2350, 2015.

[63] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," *arXiv preprint arXiv:1611.01576*, 2016.

[64] J. Howard and S. Gugger, "fastai: A layered api for deep learning," 2020.

[65] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

[66] D. Sussillo, S. D. Stavisky, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "Making brain–machine interfaces robust to future neural variability," *Nature communications*, vol. 7, p. 13749, 2016.

[67] P.-H. Tseng, N. A. Urpi, M. Lebedev, and M. Nicolelis, "Decoding movements from cortical ensemble activity using a long short-term memory recurrent network," *Neural computation*, vol. 31, no. 6, pp. 1085–1113, 2019.

[68] M. Clerc, L. Bougrain, and F. Lotte, "Brain-computer interfaces," 2016.

[69] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy, "A neural network that finds a naturalistic solution for the production of muscle activity," *Nature Neuroscience*, vol. 18, p. 1025–1033, Jun 2015.

[70] H. Morioka, A. Kanemura, J. Hirayama, M. Shikauchi, T. Ogawa, S. Ikeda, M. Kawanabe, and S. Ishii, "Learning a common dictionary for subject-transfer decoding with resting calibration," *NeuroImage*, vol. 111, pp. 167–178, 2015.

[71] G. K. Anumanchipalli, J. Chartier, and E. F. Chang, "Speech synthesis from neural decoding of spoken sentences," *Nature*, vol. 568, no. 7753, pp. 493–498, 2019.

[72] J. A. Brantley, T. P. Luu, S. Nakagome, F. Zhu, and J. L. Contreras-Vidal, "Full body mobile brain-body imaging data during unconstrained locomotion on stairs, ramps, and level ground," *Scientific data*, vol. 5, p. 180133, 2018.

[73] C. Y. Chang, S. H. Hsu, L. Pion-Tonachini, and T. P. Jung, "Evaluation of

Artifact Subspace Reconstruction for Automatic EEG Artifact Removal," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2018-July, no. June, pp. 1242–1245, 2018.

[74] S. Nakagome, T. P. Luu, J. A. Brantley, and J. L. Contreras-Vidal, "Prediction of EMG envelopes of multiple terrains over-ground walking from EEG signals using an Unscented Kalman Filter," in *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-Janua, pp. 3175–3178, 2017.

[75] J. A. Brantley, T. P. Luu, S. Nakagome, and J. L. Contreras-Vidal, "Prediction of lower-limb joint kinematics from surface EMG during overground locomotion," in *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-Janua, pp. 1705–1709, 2017.

[76] D. O. Nahmias, K. L. Kontson, D. A. Soltysik, and E. F. Civillico, "Consistency of quantitative electroencephalography features in a large clinical data set," *Journal of neural engineering*, vol. 16, no. 6, p. 066044, 2019.

[77] L. Jin, J. Chang, and E. Kim, "Eeg-based user identification using channel-wise features," in *Asian Conference on Pattern Recognition*, pp. 750–762, Springer, 2019.

[78] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, 2018.

[79] D. Wu, Y. Xu, and B. Lu, "Transfer learning for eeg-based brain-computer interfaces: A review of progresses since 2016," *arXiv preprint arXiv:2004.06286*, 2020.

[80] M. Congedo, A. Barachant, and R. Bhatia, "Riemannian geometry for eeg-based brain-computer interfaces; a primer and a review," *Brain-Computer Interfaces*, vol. 4, no. 3, pp. 155–174, 2017.

[81] O. Özdenizci, Y. Wang, T. Koike-Akino, and D. Erdoğmuş, "Learning invariant representations from eeg via adversarial inference," *IEEE Access*, vol. 8, pp. 27074–27085, 2020.

[82] C. Finn and S. Levine, "Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm," *arXiv preprint arXiv:1710.11622*, 2017.