DEVELOPING EXPLAINABLE DEEP LEARNING MODELS USING EEG FOR BRAIN MACHINE INTERFACE SYSTEMS

by

Akshay Sujatha Ravindran

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, Ph.D.
Committee Member: Pranav Parikh, Ph.D.
Committee Member: David Mayerich, Ph.D.
Committee Member: Hien Nguyen, Ph.D.
Committee Member: Cameron Buckner, Ph.D.

University of Houston
December 2021

Acknowledgements

I would like to thank my advisor, Dr. Jose L. Contreras-Vidal, without whose guidance and support throughout my Ph.D., I could not have accomplished this. I am forever grateful for all the training, faith, and opportunities that he has provided me. Secondly, I would like to thank Dr. Rose Faghih, Dr. Cameron Buckner, Dr. David Mayerich, and Dr. Hien Nguyen, for guiding me and participating on my dissertation committee. I would like to additionally thank Dr. Nguyen for making the computing cluster in his lab available for me which helped me to complete the thesis on time. I would also like to thank Dr. Charles Layne and Dr. Pranav Parikh for supporting me at different points on this journey.

A significant portion of my learning and insights came from the interactions and discussions with my colleagues in the laboratory. I owe much of the success of completion of my graduate education, and invaluable friendship and memories to all of you (Alamir Ayman, Alex Steele, Alex Craik, Dr. Andrew Paek, Chris Malaya, David Eguren, Eric Todd, Dr. Fangshi Zhu, Dr. Jesus G Cruz-Garza, Jose Gonzalez, Dr. Justin Brantley, Dr. Kevin Nathan, Dr. Manuel Cesari, Dr. Mario Ortiz Garcia, Dr. Mauricio Adolfo Ramirez, Michelle Gale, Nishant Rao, Dr. Nikunj Bhagat, Dr. Phat Luu, Dr. Rahul Goel, Ranga Prasad, Dr. Sho Nakagome, Dr. Subasree Ramakrishnan, Dr. Yongtian He, Zach Hernandez). I would like to specifically acknowledge Dr. Justin Brantley, Dr. Andrew Paek, and Alex Craik who have always gone over and above to provide great mentoring, and support, both professionally and personally during different stages of my doctoral study.

None of this would have come through without the opportunity given to me by Mr. Preejith S.P. and Dr. Mohanasankar Sivaprakasam at HTIC, IIT Madras, who took me in and decided to give me a chance to work at the research center. The training there was pivotal in shaping my research interest. Additionally, I also want to thank Dr. Muhammed Shanir PP and Prof. Sunitha Beevi at the TKM College of Engineering for their confidence in my abilities even during the lowest points of my

academic journey. Lastly, I want to thank my family back in India for their never-ending faith in me and for always facilitating the journey to my goals and dreams. Also, I could not have gone through these challenging yet fruitful period in my life without the constant support, love and care of my wife. I would like to thank you for standing by me through all the busy and hectic years we had. Thank you for being both my strongest cheerleader and anchor!

This research was supported by the NSF IUCRC Building Reliable Advances and Innovation in Neurotechnology (BRAIN) Center Award (NSF Award 1650536). I am also grateful for the support of the Core facility and resources provided by the Research Computing Data Core at the University of Houston which was also critical to completing the work in a timely manner.

Abstract

Deep learning (DL) based decoders for Brain-Computer-Interfaces (BCI) using Electroencephalography (EEG) have gained immense popularity recently. However, the interpretability of DL models remains an under-explored area. This thesis aims to develop and validate computational neuroscience approaches to make DL models more robust and explainable. First, a simulation framework was developed to evaluate the robustness and sensitivity of twelve back-propagation-based visualization methods. Comparing to ground truth features, after randomizing model weights and labels, multiple methods had reliability issues: e.g., the gradient approach, which is the most used visualization technique in EEG, was not class or model-specific. Overall, DeepLift was the most reliable and robust method. Second, we demonstrated how model explanations combined with a clustering approach can be used to complement the analysis of DL models applied to measured EEG in three tasks. In the first task, DeepLift identified the EEG spatial patterns associated with hand motor imagery in a data-driven manner from a database of 54 individuals. Explanations identified different strategies used by individuals and exposed the issues in limiting decoding to the sensorimotor channels. The clustering approach improved the decoding in high-performing subjects. In the second task, we used GradCAM to explain the Convolutional Neural Network's (CNN) decision associated with detecting balance perturbations while wearing an exoskeleton, deployable for fall prevention. Perturbation evoked potentials (PEP) in EEG ($\sim$75 ms) preceded both the peak in electromyography ($\sim$180 ms) and the center of pressure ($\sim$350 ms). Explanation showed that the model utilized electro-cortical components in the PEP and was not driven by artifacts. Explanations aligned with dynamic functional connectivity measures and prior studies supporting the feasibility of using BCI-exoskeleton systems for fall prevention. In the third task, the susceptibility of DL models to eyeblink artifacts was evaluated. The frequent presence of blinks (in 50% trials or more), whether they bias a particular class or not, leads to a significant difference in decoding when using

CNN. In conclusion, the thesis contributes towards improving the BCI decoders using DL models by using model explanation approaches. Specific recommendations and best practices for the use of back-propagation-based visualization methods for BCI decoder design are discussed.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background

1.1 Brain-Machine Interface Systems

Brain-Computer Interface (BCI) systems provide means by which one could use the brain activity measured either invasively or noninvasively to interact with an external device or their environment [1]. These systems record the brain activity, process the signal, and translate relevant features into commands which drive an end-effector that can be used to control a virtual or physical machine such as a computer, robot, exoskeleton, prosthetic, or even a digital avatar [2]. BCI systems are being used in both assistive modes such as providing means for individuals who are paralyzed to control external devices/communicate or as a rehabilitation tool to improve their recovery process [3]. BCI systems have also proved to be useful in assisting individuals with different neuromuscular and neurological disorders such as spinal cord injury [4], stroke [5], cerebral palsy [6], etc. BCI can compensate, restore or replace their reduced functional capabilities and facilitate neural recovery.

A typical BCI system contains multiple stages of pre and post-processing indicated by Fig 1.1. The artifact removal stage contains different pre-processing steps which handle most of the artifacts that contaminate the brain signals. This is usually followed by a feature engineering stage wherein the most relevant features for the particular task of interest are identified. These features are then used to train a classifier/regression model to generate the commands for controlling an external device [7].

Recent advancements in machine learning and deep learning-based decoders have

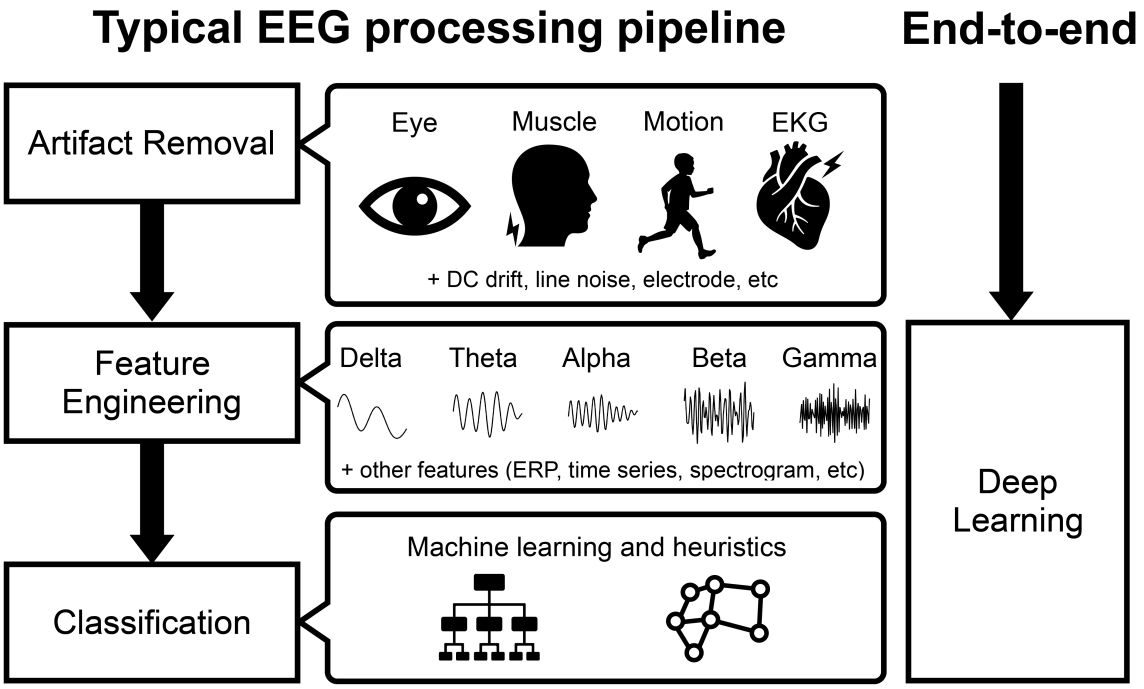

Figure 1.1: Example processing pipeline used to develop EEG-based BMI.

led to significant improvement in decoding capabilities using EEG. Lately, with the advancements in deep learning (DL), studies adopting such models as decoders have exponentially increased. DL models use a computational framework that has multiple layers that learn representations at multiple levels of abstraction [4]. In addition to improving the predictive power, the utility of DL is mainly inspired by the possibility of removing this multi-stage processing of EEG. Many studies have been using deep learning models to function in an end-to-end manner wherein the same model is supposed to handle the artifacts, identify relevant features, as well as perform decoding [8] [9]. Over 60-70 % of studies do not handle artifacts when using deep learning models. [8] [10] [9]. The possibility of not handcrafting the features required for decoding is also an advantage of using DL models. The model would be able to automatically identify the relevant features thus not limiting the decoding to the hand-picked or pre-selected features. A review by Roy et al. [9] reported that studies

Table 1.1: Different aspects illustrating the importance of explainability of deep learning models.

Sl No	Explainability aspect	Factors and applications
1	Model debugging	Failure mode analysis Assessing the impact of artifacts Improving model performance
2	Regulatory oversight	FDA jurisdiction Compliance Trustworthiness
3	Scientific insights	Data-driven feature learning Learning from the state of the art models
4	Adoption	Automation bias Model skepticism
5	Ethics	Transparency Decision-making accountability
6	Bias	Training data not representative of the population Proxy measures as the dependent variable

have reported a median decoding increase of 5.4% between DL algorithms and traditional ML algorithms demonstrating the benefit of using DL models as a decoder. However, these models do suffer from poor interpretability and explainability which limits their widespread adoption in spite of the performance improvement, especially in industries such as healthcare [11] [12].

1.1.1 Why is explainability important?

The implications and the need for explaining the model decision are multidimensional. Some of the major aspects are summarized in table 1.1

Using explainability approaches allows us to understand the failure modes in the model, giving valuable insights about the model. These methods will help debug the model by identifying some of its limitations and mistakes thereby improving the model. When the DL models are translated to EEG, some key challenges need to be addressed. Even though EEG provides significant advantages over other measurement modalities, one of the biggest limitations of EEG is the relatively lower SNR.

Many of the artifacts of physiological and non-physiological origin such as eye blink/ movement, muscular artifacts, cable pops, etc typically have much higher amplitude compared to true brain signals. There exists a possibility that the models could be learning from these artifacts and not the real brain signals. The reliance on spurious correlations present in the data is commonly referred to as the "Clever Hans Problem" [13]. Therefore, it is essential to understand the decision-making process to know why a model arrived at a particular decision. A model learning spurious correlations will fail when deployed in the real-world making it useless. Using explainability approaches can help debug these potential confounds.

From a regulatory standpoint, if an artificial intelligence software making a recommendation to the healthcare professional is not explainable, the software will fall under the FDA jurisdiction and will have to clear stringent long regulatory pathways for its usage [14]. On the other hand, explainable recommendations to healthcare professionals do not fall under the regulatory powers of the FDA [14]. When developing software solutions, from a deployment perspective, if the solution does not cause significant risk and if the regulatory oversight could be avoided, it will be a critical design consideration, to avoid regulatory complications as they can be time and resource-intensive. A significant emphasis on the explainability of DL models is required as FDA is shifting towards expanding the regulatory oversight to artificial intelligence and machine learning software, most of which were exempt under the Cures Act [15]. Currently, the FDA is developing an action plan on how best to regulate AI/ML-based software [16]. Similarly, under the newly proposed EU general data protection regulation, the new law creates a "right to explanation", whereby the user can ask for the explanations of an algorithmic decision that was made about them [17]. With various regulatory bodies expanding their jurisdiction to regulating algorithmic decision making, knowing whether the DL solutions would be compliant

with such regulatory requirements would be an important aspect going forward. Additionally, having explainable decisions from the DL models also helps to improve the trust in the model.

One of the advantages of using DL is the possibility it offers to avoid the need for hand crafting features. The model can automatically identify relevant patterns required for decoding. This is another important area in which explainability would offer tremendous possibilities. Understanding what the models are looking at could lead to new scientific discoveries and progress the field forward [18] [19]. Recent studies are also attempting to better understand how different state-of-the-art models are beating human experts, E.g. study evaluating how Alpha Zero beat human experts at chess [20].

The popularity of deep learning is not without its share of skepticism. In terms of the adoption of DL models, there exist two subsets of people at either end of the spectrum. The first category puts too much trust in the AI model assuming it is error-proof while the other group consists of skeptics who are hesitant to adopt [21]. Often, many experienced researchers tend to fall into the latter category since they are uncomfortable using a model without understanding how the model arrived at a particular decision. The automation bias can be detrimental too particularly in applications aimed at helping the clinical population or high-risk applications. Consider a model for decoding motor imagery for stroke rehabilitation purposes. Ideally, a model should learn neural features from regions of the brain which has representations of limb movement (typically motor-related signals). Say a DL model exhibits significant performance gain compared to traditional models, but if the model is learning from irrelevant noise signals instead of motor-related potentials, the rehabilitation will not

be effective and the high-performance increase becomes insignificant. Ideally, the predictive models should utilize neural features associated with the task-specific region to induce therapeutic plasticity, rather than an unrelated neural activity that is not associated with the motor task. Similarly, if explanations could be provided on what the model was looking for when making the decisions, the people who are hesitant to use these models could be more open to their adoption.

Ethics is yet another major aspect in which explainability will play a huge role. There is a growing popularity for devices in the consumer industry that make use of EEG for different applications including sleep monitoring [22], neurofeedback applications [23], measuring attention and fatigue levels [24]. Some of these devices have already garnered their fair share of criticism and concerns regarding their implementation. Recently, Brainco [25], a consumer neurotech company came under scrutiny. Their system used a dry EEG system and use proprietary algorithms to measure whether students are attentive or not and provides a report to faculty and parents. Should one believe the device that gives a number without giving a proper explanation or the student if they say they are being attentive? By not being aware of how the attention scores are estimated or providing means to debug whether artifacts are biasing these systems, the product providing such single-number metrics can cause tremendous pressure and ethical challenges to both students, teachers, and parents. The cause of concern related to privacy and psychological stress resulting from wearing these devices led to the government suspending their use [26]. With more consumer companies moving towards using novel machine learning and deep learning models in their applications, explainability will be key in building trust and could help in addressing some of the ethical dilemmas caused by the use of such systems.

Similarly, knowing whether the model is discriminative towards any particular community due to bias present either in the training data or how the training was

performed will be critical when the solutions get deployed [27]. If the training data poorly represents the entire population, this could lead to biases in its decision-making process. In a recent study, it was identified that multiple datasets used for facial recognition had an overwhelming majority of lighter-skinned subjects resulting in reduced accuracy to detect darker-skinned faces [28]. A similar racial bias was identified for prediction algorithms used in health care [27]. They identified that black patients who are assigned the same level of risk by the algorithm were found to be sicker than the white patients. The bias was caused by the algorithm using health cost as a proxy for health needs. Interpreting how and why a model is arriving at specific decisions will be critical to eliminate similar biases in algorithms.

1.1.2 Model Explanation approaches

Even though there exist many variants of the algorithms being developed to interpret the neural network models, the broad majority of them could be categorized into three categories: Model Distillation, Visualization methods, and Intrinsic methods [29]. A summary of the different types of model explanations is given in Fig.1.2.

Distillation/Approximations Methods

A group of approaches tries to approximate the DL models with simpler models whose input-output behavior mimics that of the DL model. Later, by interpreting the simpler model, insights into how the complex model works can be obtained. These approaches are broadly labeled under the category of distillation methods. One of the most popular among these methods would be the use of the Local Interpretable Model-agnostic Explanations (LIME) method [30].

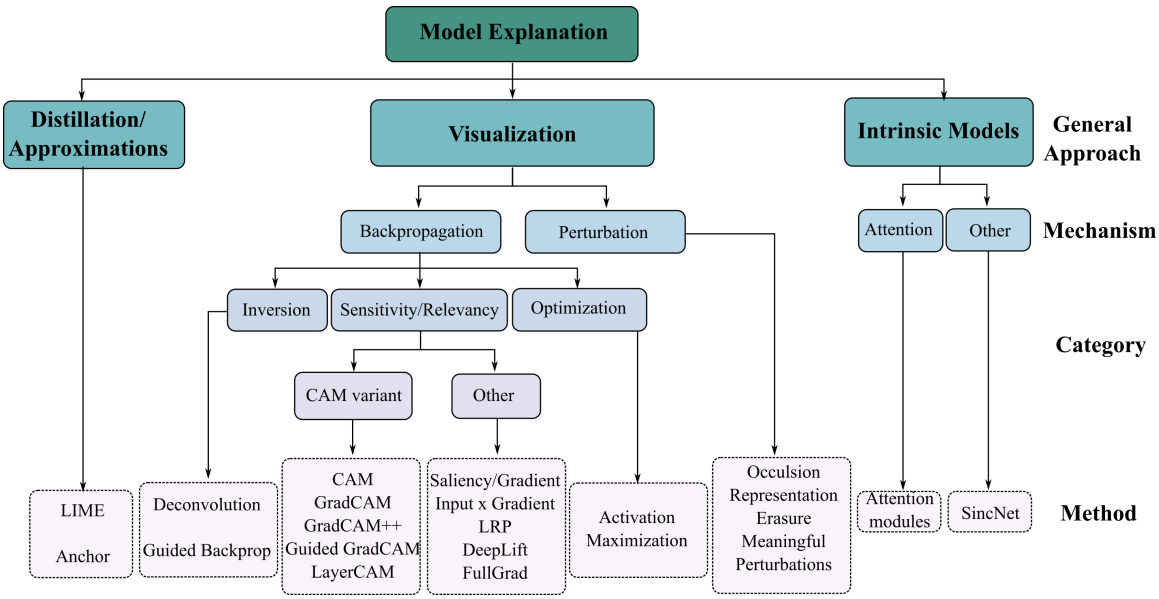

Figure 1.2: Different types of explanation approaches in neural network models.

Visualization

Visualization methods are approaches which in general highlight the most important feature or attribute present in the input that affects the decision of the model through different visualization. One of the most common approaches is the saliency maps which highlight the important segment of the input. These could further be divided into different categories based on how they are implemented. The majority of the approaches developed in this category are based on back-propagation [31]. The gradient/relevancy score for a particular class or neuron is back-propagated in some form for these approaches. The most common and oldest approach is the Gradient approach [32] which is estimating the gradient of the output with respect to the input. Variants of the simpler models have been developed which are more robust and less noisy like FullGrad [33], Input X Gradient, Layerwise Relevance Propagation [34], DeepLift [35] or different approaches of class activation maps likes GradCAM [36], GradCAM++[37], LayerCAM [38], GuidedGradCAM [39], ScoreCAM [40] etc. There are a few methods that attempt to reverse the forward operations ('Inversion') in a

CNN such as Deconvolution [41] and Guided Backpropagation [42]. Other approaches like activation maximization involve adding an additional 'optimization' step wherein it tries to create an input the maximizes the score for a particular class/ filter of interest [32]. Through all of these methods, the researcher gets additional context through different ways of scientific visualization on what drives a model decision.

Intrinsic Methods

Intrinsic methods involve either developing models which provide an explanation for the decision as part of its model output or those in which explanations can be extracted from the architecture rather straightforward way [29]. Some common methods involve models using the attention mechanism [43]. The attention mechanism generates a contextual vector for downstream processing by learning a conditional distribution over the input. Some studies on the other hand engineer the deep network to perform specific meaningful functions which are easily interpretable. One such approach is the development of SincNet [44] which is based on parameterized sinc functions wherein the model learns cutoff frequencies for the filter banks. This allows for more easily interpretable filters as the most highly activated units would correspond to a particular frequency band.

1.1.3 Scope

In the EEG literature, a majority of the model explanations are based on the visualization method using the backpropagation approach. The scope of the dissertation will be limited to the visualization approach as this is also the most extensively developed explainability method in other domains as well [45] [46]. Limiting the scope to these methods further allows for a more straightforward comparison of their effectiveness. The dissertation is aimed at finding some best practices to adopt different visualization-based model explanations methods for EEG applications.

For the first specific aim (SA1), multiple visualization-based explanation methods would be tested on simulated EEG to understand the ground truth sensitivity and robustness of these methods. This would help understand when these approaches fail and identify the most suitable method for EEG. Simulated data allows isolation of distinct EEG features such that only the particular feature of interest would be different between the classes and can produce a selective and controlled variation of these features. This will help with providing a more objective assessment of the robustness and sensitivity of these approaches to different features.

Next, for specific aim 2 (SA2), understanding from simulated data would be translated to real EEG data, to evaluate whether certain DL models can discover the underlying brain dynamics in a data-driven manner and identify if these methods can give insights into whether the model is biased by artifact or not.

Next, for specific aim 3 (SA3), these identified explainability tools and the simulation framework would be used for model debugging purposes to explore the influence of eye blink artifacts. To summarize, the following theme of questions would be covered in the dissertation:

1. Which visualization based model explanation approaches are more suitable for EEG? (SA1: Chapter 2)

2. Can model explanations provide neurological insights into the underlying brain dynamics during different BCI paradigms? (SA2: Chapter 3a, 3b)

3. Instead of looking at individual explanations, can the model explanations be aggregated to learn global, class-specific patterns? Can this framework be used to assist with model debugging? (SA2, SA3: Chapter 3, 4)

4. Will the DL model learn to avoid eye blink artifacts when trained in an end-to-end manner without manually removing them? If not, how would the model be influenced by eye blinks? (SA 3)

Chapter 2

An Empirical Comparison of Deep Learning Explainability Approaches for EEG using Simulated Ground Truth

2.1 Introduction

With the popularity of neural networks in recent years, the field of deep learning has gained exponential growth in the last decade. They have become the state of the art model in different domains including computer vision [47], natural language processing [48],[49], etc. They started beating human performance in many tasks such as the game of GO [50], and recently solved 50-year-old grand challenge of protein folding problem [51]. Even in EEG, they have shown a median improvement of 5.4% classification score in various applications [9]. There exists concern on whether this improvement in decoding is from learning the underlying true data distribution or learning spurious artifacts present in the data [13],[52],[53].

The emphasis on explainability hasn't picked up a similar pace in popularity compared to deep learning in general for EEG applications. The adoption of explainability for deep learning models in the research involving EEG is still very rare. To better quantify the number of studies that employ explainability approaches when using deep learning on EEG, a literature review was conducted using the Web of Science. The advanced search option was used with the criterion ((AB=(EEG) OR AB=(Electroencephalography)) AND (AB= (neural network) OR AB= (deep

learning) OR AB = (CNN) OR AB = (Convolutional Neural Network) OR AB = (Recurrent Neural Network) OR AB = (LSTM) OR AB = (GRU))) AND (ALL=(interpretability) OR ALL=(explainability) OR ALL=(interpretable)). The search conducted in November 2021 gave a total of 65 publications. Among these 30 did not use any specific explainability method in the paper. They either only refer interpretability/ explainability in the paper for discussion purpose or is not relevant. A few of the papers that include interpretability in title/abstract used hand-crafted features to train the model and refer to them as "interpretable models". These studies were also not included. Two papers were not considered because of poor quality. After removing these papers, only 33 studies remained that used some form of model explanation. On the other hand, studies without the part (ALL=(interpretability) OR ALL=(explainability) OR ALL=(interpretable)) in the advanced search provided a total of 5,951 papers suggesting the studies including model explanation currently is less than 0.6 %.

The types of methods used in the 33 studies is summarized in Fig. 2.1. The majority of the studies use some form of heatmap approach. These heatmap approaches highlight the part of the input data the model is looking at to arrive at the correct prediction. The most commonly method (Saliency) is also the most simplest wherein the gradient w.r.t. input was computed [54],[55],[56],[57],[58],[59],[60],[57],[61]. The next commonly used method is plotting the convolutional filters directly; usually, the convolutional filters that have a kernel spanning the entire EEG channels (spatial convolutional layer weights) [62],[63],[64],[65]. However, looking at the raw weights does not directly indicate whether they are class-specific features or not. Considering there is a large number of filters, the ideal combination of filters that contribute positively to the prediction would be difficult to discern. Also, previous studies have shown that significant non-zero weights can be observed for channels whose activity can be independent of the underlying cortical activity [66]. Many other studies used occlusion based model explanations wherein they occlude or zero out parts of the

input to identify the most sensitive region. However, occlusion methods are not ideal when there are dependencies between non-local features. In that case it has to be known apriori how to define the mask to include these dependencies (width, shape of mask, etc). Other studies have used more complex versions of back-propagation approaches. E.g. Sturn et al. (2016) used LayerWise Relevance Propagation (LRP) to identify scalp relevancy associated with motor imagery [67]. Similarly, Lawhern et al. (2018) used the Deep Learning Important FeaTures (DeepLift) method [65] for motor imagery and error-related negativity response task. Ravindran et al. used GradCAM to demonstrate that CNN was learning from common perturbation evoked potentials in single-trial EEG [68]. A good number of studies used the activation maximization approach [32] to synthetically generate inputs that maximally activate a particular neuron, typically the final layer neurons [69],[70],[71]. Few studies attempted a perturbation approach in which they perturb the input and evaluate the change in output [72],[73]. The other category includes studies that use approaches not commonly used. Most of them either visualize clustering of hidden layer activation to show class separation [74] or show a correlation of hidden layer activation to different features [75],[76].

Recent research in computer vision has shown that many of these visualization-based approaches when applied to images have reliability issues [77],[78]. Adebayo et al. (2018) showed that visual inspection of model explanations alone can mislead into giving compelling cases. They demonstrated that many of the commonly used explainable methods lack sensitivity to the model and the data generating process [78]. In that study, they randomized the labels and separately reinitialized the model weights. Then they hypothesized that if the model was specific to data / the trained model the explanations should be significantly different with randomization. However, they found that many methods were invariant to these manipulations and only gradients and GradCAM passed their sanity checks. In a separate study Kindermans et al. (2019) show that many methods do not satisfy input invariance either [77].

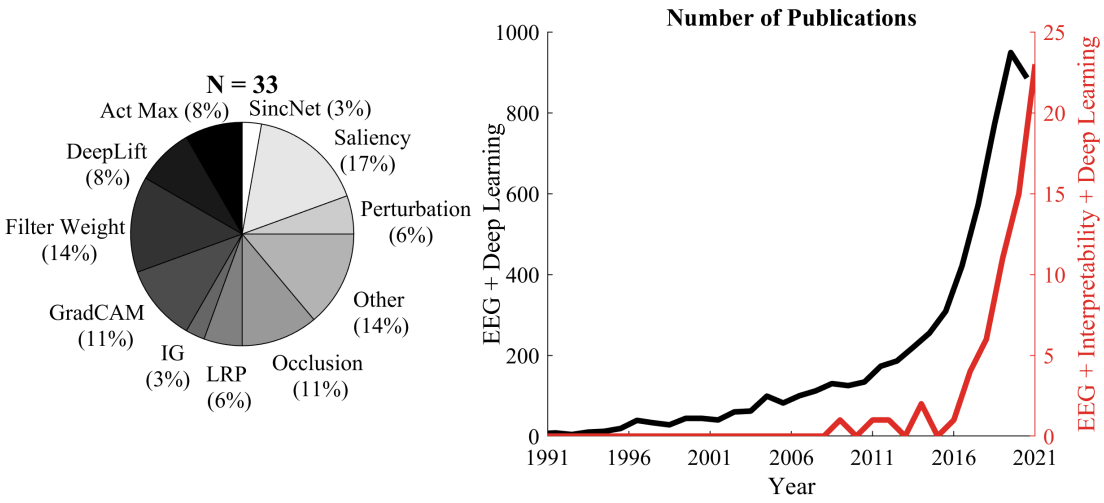

Figure 2.1: Left: Pie chart showing the distribution of methods used in the screened studies from the web of science search. Right: Trend showing the number of EEG publications using deep learning, with and without explainability (not screened).

Most of these studies in EEG limit visualization to either one example or an average of one subject. Thus, it is not clear whether the proposed methods would generalize to other datasets. Therefore, it remains unclear which explainability method(s) are robust and reliable when applied to EEG data, and whether or not these methods are sensitive to only certain features in EEG. The sensitivity element is equally important on top of robustness because unlike images, EEG is a bit more complex with features in multiple domains such as temporal, spectral, and spatial domains all equally relevant. Looking at raw time series is less intuitive relative to looking at an image. Also, finding the ground truth in real EEG is a challenging task particularly with the lower values of SNR. Even the same task repeated might have a large source of variability due to the nature of how the human brain works, the influence of the environment, etc. Knowing the exact location of a particular feature in time could be difficult to ascertain when looking at individual trials as well. In addition, often multiple features and noise superimpose making it difficult to know which feature the model is sensitive to. Here, the sensitivity and robustness of 12 of the heatmap based

methods for different sources of signal with varying signal-to-noise ratios (SNR) were evaluated. The methods will be evaluated to know if they can identify the ground-truth signal accurately as well as whether the explanations are both class specific and model specific. This research proposes to compare the strengths and weaknesses of different methods to better understand the pitfalls and provide recommendations for the appropriate application.

2.2 Methods

2.2.1 Convolutional Neural Network

The architecture for the model is summarized in Fig. 2.2. The intention was to use a very generic CNN model without any specialized architectural changes. This was done to ensure generalizability to existing studies. The input to the model is the 1 s EEG window (batch size $\times$ 250 samples $\times$ 54 channels). Eight channels were removed as they are not contained in the forward model. The model consisted of 5 temporal convolution layers of 32 units each (5×1 kernel size with a stride length of 1) and 1 spatial convolution layer of 32 units (1×62 kernel size). The number of convolutional layers was kept as 6 as the majority of the prior studies used 6 or lower convolutional layers [8]. This also aligns with how the motor cortex is arranged, which is organized into a total of 6 layers as well [79],[80]. We would like to emphasize here that having a similar number of layers does not necessarily enforce that each layer would replicate the actions of each layer of the motor cortex. The filter size was selected such that the total receptive field for the final convolutional block would span at least half the sampling rate (125 Hz). A temporal pooling layer of 2×1 pooling dimension with a stride length of 2 was also used after every convolutional filter layer except the last two blocks. The output from these convolutional layers was flattened and fed into a dense, fully connected layer of 32 hidden units followed by an output layer with

softmax activation.

A dropout layer with alpha = 0.5 was added in between the dense layer and the output layer to reduce overfitting. Except for the output layer, the model utilized ReLU as the activation function. ReLU was used as the activation function as this was also the most popular activation function used (70% of studies [8]). The proposed model was implemented in python 3.7 using Pytorch library [81]. For each of the condition (temporal, spatial and spectral), an independent model was trained to classify the distinct classes. A 5-fold cross-validation was performed and model explanations and the comparison metrics were estimated on the test set from each fold. The value across the folds are then compared between the type of model explanations.

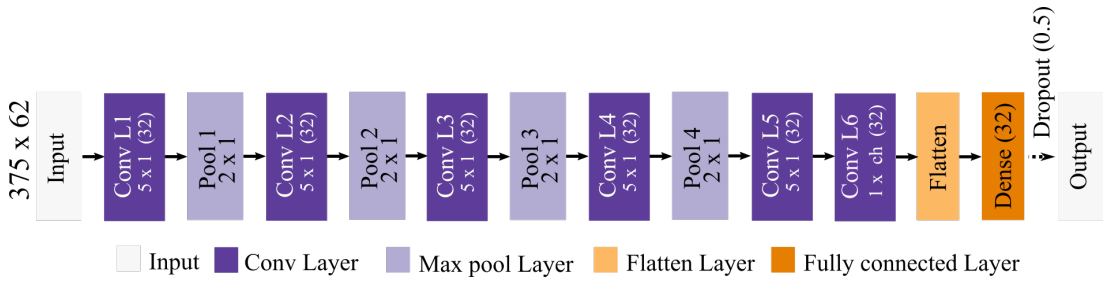

Figure 2.2: Model architecture: Each block correspond to different types of layers in the model. The dotted line illustrates the dropout operation during the training phase aimed at reducing overfit. During inference, all units were retained.

2.2.2 Simulated Data

To compare the relative performance of different model explanation methods, the SEREEGA library [82] was used to simulate ground truth EEG features. The typical workflow used to simulate EEG activity using SEREEGA is summarized in Fig 2.3. The process starts by defining the lead field matrix and the head model. The New York head model was used for generating the lead field matrix [83]. The toolbox supports the pre-generated leadfield that includes 75,000 source locations which could be projected to 228 sensor locations on the scalp. The New York head model does

detailed segmentation of six types of tissues (scalp, skull, cerebrospinal fluid, gray matter, white matter, air cavities). Later, the source location was selected to project the feature from. The source location could either be randomly selected or chosen manually based on the Montreal Neurological Institute (MNI) coordinates [84]. Later, the orientation for the dipoles was chosen. Each source has a default orientation associated with it. But, the orientation that is either tangential or perpendicular to the scalp for each of the dipoles can also be chosen. For this study, all dipoles are chosen to be perpendicular to the scalp surface to improve the localization of the scalp projection for ground truth.

Once the source and the orientation are selected, an activation/signal would be added to these sources. SEREEGA offers systematic deflections in the time domain to simulate event-related potentials as well as systematic modulations of oscillatory activity to simulate event-related spectral perturbation. The toolbox also allows simulating different types of additive noises (pink, white, brown, etc). Once the appropriate signal and noise are added, it allows mixing the signal and noise in varying proportions such that different combinations of Signal-to-Noise Ratio (SNR) could be achieved at the projected scalp EEG. In addition, uncorrelated white noise was added to simulate sensor noise. Using the combination of signal, noise, source location and orientation the toolbox allows creating ground truth simulated EEG with varying localization capabilities in temporal, spatial, and spectral domains.

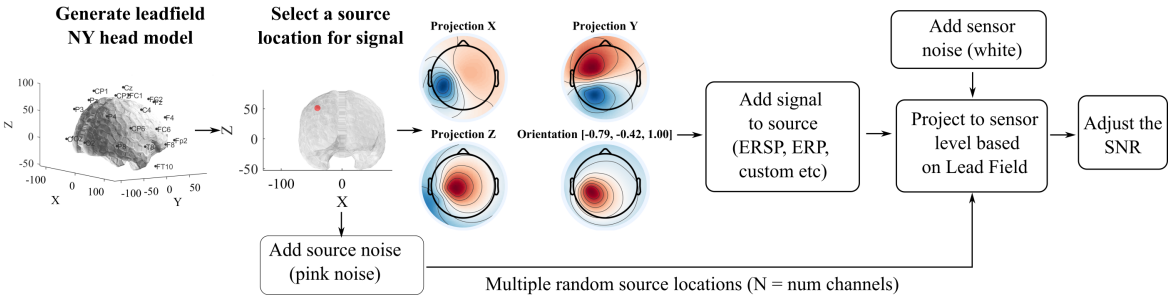

Figure 2.3: Steps present in generating different types of features in simulated EEG using the SEREEGA toolbox.

For all the simulations, the leadfield matrix with projected on to actiCAP64 channel configuration from the sources was used. The sampling rate was set to 250 Hz with the window size of each simulated epoch to 1 second long. To replicate brain noise, sources equaling the number of channels - the number of signal dipoles were uniformly selected randomly across the brain surface and a 5μV pink noise were added to these sources similar to the simulation replication done by Krol et al. 2018 [82]. For each condition, to evaluate the performance impact under varying SNR, the noise was added to yield the following SNR: -3.5 dB, -12 dB, -16 dB, -19 dB, -23 dB. Fig 2.4 shows an example of the difference when the simulated ERP component gets added with noise at varying SNR.

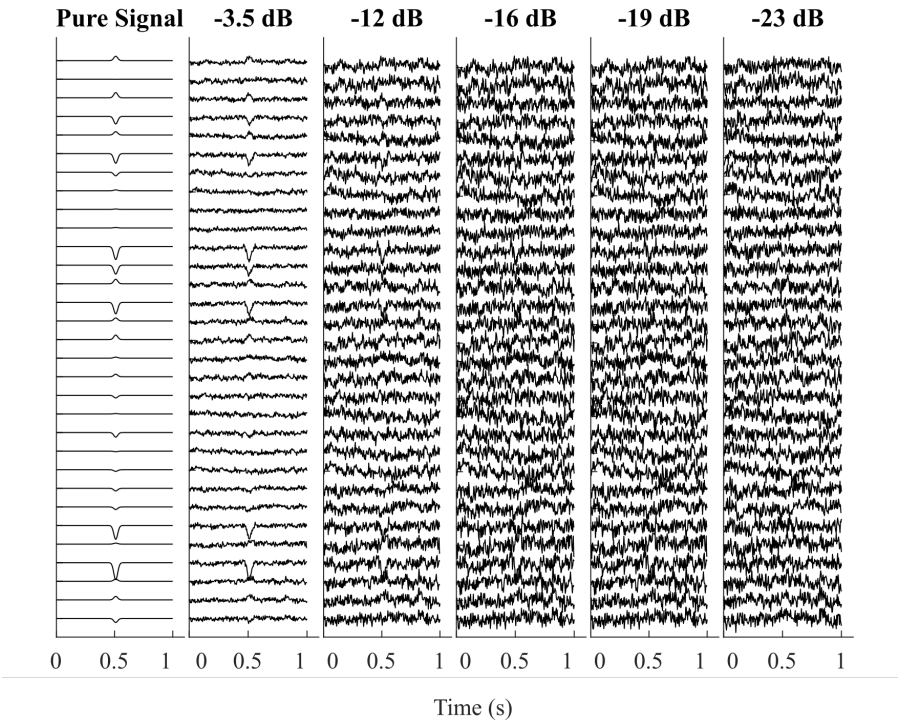

Figure 2.4: Representative example to demonstrate the effect of varying SNR on an ERP component.

Event-Related Potential Components

To evaluate how different model explanations fair in localizing the temporal aspect of EEG, different ERP components were simulated. Four distinct classes of ERP components were simulated with N = 10000 per class. For each epoch, the source location was sampled from one among 10 source locations in Table 2.1. Even though the precise location is not very important, in order to have some constraint, source locations were selected corresponding to perturbation evoked potentials based on ranges suggested in the source analysis results from prior studies [85],[86],[87]. These components were selected as the balance perturbation task in specific aim 2 elicits perturbation evoked components. The source locations in the MNI coordinates are shown in Fig2.8.

The following attributes for the source components were tested for in the simulation.

1. Class 1: Time locked positive deflection of EEG. Class 1 contained a positive component centered at 60 ms with 8 s.d. latency with a peak width of 50 ms ± 2. The amplitude of the component was randomly sampled between 1 µV to 13 µV uniformly. The component's magnitude and width closely resemble the characteristic range of the P1 component in perturbation evoked potentials [88]. One among the first 5 source locations from Table 2.1 was selected randomly as the source location.

2. Class 2: Same properties as Class 1 but different latency (latency difference). Class 2 contained a positive component centered at 900 ms ± 5 s.d. latency with a peak width of 100 ± 4 ms. The amplitude of the component was the same as that of Class 1. However, latencies were shifted to avoid overlap between the two classes to better quantify and compare the explainability techniques. One among the first 5 source location from Table 2.1 was selected randomly as the

Table 2.1: MNI coordinates of the ERP sources.

No.	Dipole Location	MNI coordinates		
		x	y	z
1	Paracentral lobule	-9.1	-8.5	60.2
2	Paracentral lobule	10.1	-6.9	62.3
3	Paracentral lobule	4.6	-3.4	54.3
4	Paracentral lobule	8.4	-9.9	57.9
5	Posterior cingulate	7.5	-1.6	53.5
6	Precuneus	-2.6	-33.9	54.5
7	Posterior cingulate	-3.5	-30.7	52.1
8	Precuneus	-4.1	-43.2	49.7
9	Isthmus cingulate	-3.6	-39.2	46.1
10	Posterior cingulate	-3.3	-26	50.4

source location.

3. Class 3: Same magnitude as Class 1 and 2 but negative deflection instead of positive (sign difference). Class 3 consisted of an ERP component with the same amplitude of class 2 but inverted with a latency centered at 500 ms ± 8 s.d. and a width of 100 ms ± 4 s.d. One among the first 5 source location from Table 2.1 was selected randomly as the source location.

4. Class 4: Same magnitude and sign as class 3 but a different source location (source difference). Class 4 consisted of a signal of the same properties as Class 3 except that the source location is different. One among the source location (6-10) from Table 2.1 was selected randomly as the source location.

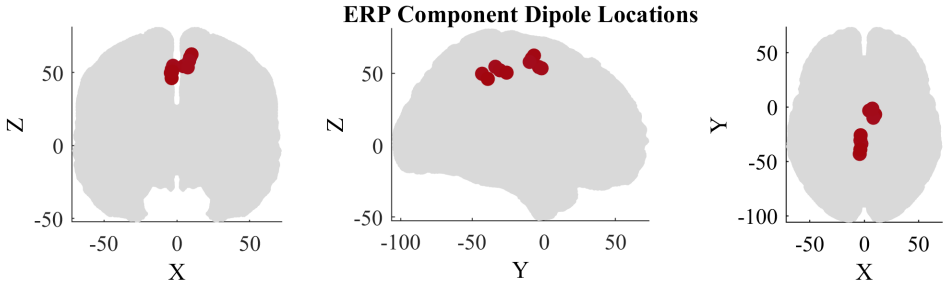

Figure 2.5: Dipole locations in MNI coordinates for the ERP components.

Spectral Perturbations

To test the sensitivity to detect spectral perturbation events, four separate classes of data were simulated each belonging to spectral perturbation events happening in four separate frequency bands. The magnitude of the signal was set to 0.5-3 μV [82]. For each epoch, the magnitude and the latency were kept the same for all classes and they only differed in their spectral content/ frequency. The latency of the center of the spectral burst for each epoch was uniformly random sampled to be between 200 and 500 ms to add a source of variability. The burst width was randomly sampled to be between 400 ms and 600m. The MNI coordinates used for the sources are summarized in table 2.2. The source location was referenced based on dipoles associated with motor imagery/execution from prior literature [89],[90],[91]. For each epoch, one of the dipole locations was selected at random to act as the source. All the dipole locations are shown in Fig 2.6.

1. Class 1: Spectral perturbation in the frequency band of 3-8 Hz. The magnitude, latency, and width of the burst were randomized between epochs.

2. Class 2: Spectral perturbation in the frequency band of 8-13 Hz. The magnitude, latency, and width of the burst were randomized between epochs.

3. Class 3: Spectral perturbation in the frequency band of 14-30 Hz. The magnitude, latency, and width of the burst were randomized between epochs.

4. Class 4: Spectral perturbation in the frequency band of 30-58 Hz. The magnitude, latency, and width of the burst were randomized between epochs.

The representative example of simulated EEG from each of the classes is shown in fig 2.7.

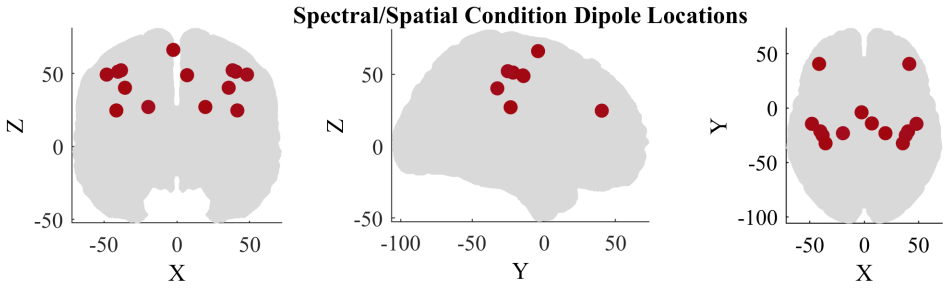

Figure 2.6: Dipole locations in MNI coordinates for both the spectral and spatial conditions.

Table 2.2: MNI coordinates of the dipoles selected for the spectral perturbation and spatial condition simulations.

No.	**Dipole Location**	**MNI coordinates**		
		x	**y**	**z**
1	L Superioparietal	-40	-21	51
2	R Postcentral gyrus	40	-21	51
3	L Superioparietal	-38	-26	53
4	R Postcentral gyrus	38	-26	53
5	L Postcentral gyrus	-48	-15	50
6	R PostCentral gyrus	48	-15	50
7	L Cingulate gyrus	-24	-24	32
8	R Cingulate gyrus	24	-24	32
9	L Supramarginal gyrus	-34	-32	38
10	R Superior parietal	34	-32	38
11	L Rostral middle frontal gyrus	-42	40	25
12	R Caudal middle frontal	42	40	25
13	L Paracentral	0	-4	65
14	R Posterior cingulate	8	-12	52

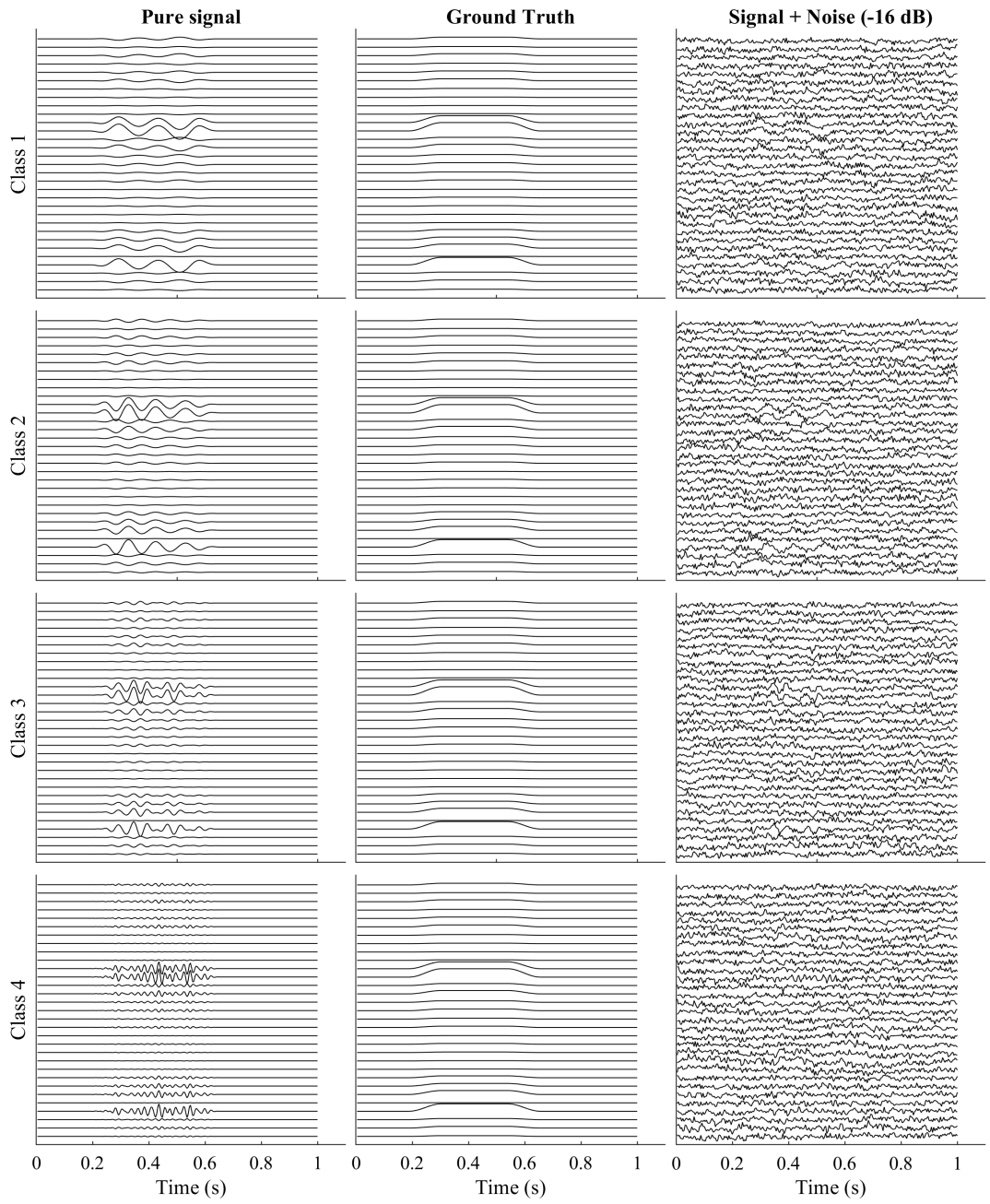

Figure 2.7: Example simulated data waveforms for each of the classes of spectral perturbations. For visualization purposes, every other channel from the true signal, ground truth, and the signal+noise waveform from one epoch is shown.

Spatial Precision

Different ERP components and ERSP perturbations with identical properties but different dipole location was simulated to assess the channel specificity. The only separation between the two classes created here is the location of the source signal. Class 1 had dipoles localized in the left hemisphere and Class 2 contains dipoles in the right hemisphere. Here the model is expected to learn all the distinct features and localize the correct scalp projection. The dipole source location for Class 1 was randomly selected from all source locations in the left hemisphere in table 2.2. Class 2 on the other hand corresponds to locations in the right hemisphere in table 2.2

2.2.3 Robustness and Sensitivity Analysis

For each condition, the simulated EEG with the respective properties are generated as discussed before. This signal is then forward projected. Noise is later added with varying levels of signal-to-noise ratios as discussed before. To get the ground truth explanation, the tapered window corresponding to the signal location was forward projected using the same lead field matrix. The segment outside of the projected signal would have a value of 0. The section with the signal (across all the channels) was normalized by dividing by the maximum value. The sensitivity/accuracy of each method was compared by evaluating the performance metrics (discussed below) w.r.t. this ground truth data.

To test the robustness of each of the explanation methods, the approach used in Adebayo et al.2018 [78] was adopted. Once the original explanation was obtained, the explanation after independently randomizing the labels and the model weights was re-computed. This tests whether the explanations are class or model-specific. The similarity of explanations w.r.t. the original explanation based on the absolute Pearson's correlation coefficient and the SSIM measure (detailed later) was estimated.

Ideally, if the model is accurate, it should have high similarity to the ground truth. On the other hand, if explanations are model and/or class-specific, the randomization performed should yield very dissimilar explanations to the original explanations. If the explanations are very similar even after randomizing, it indicates that the explanation is not very robust. The process was repeated for each type of signal/condition and SNR levels for all the explanation methods being compared.

2.2.4 Explanation Methods

The different types of visualization-based explanation methods being compared in this study are detailed below. All the methods were implemented in Python using Pytorch 1.7.0 framework [81] using either Captum 0.4.0 [92] or the Pytorch-grad-cam toolbox [93].

Gradient/Saliency (S)

Gradient or basic Saliency map as referred to in some studies is probably one of the earliest yet commonly used model explanation approaches. The gradient gives a measure of how a change in input x would change the prediction $S(x)$ in a small neighborhood around the input [32]. It is given by

$$Saliency/Gradient = \frac{\partial S}{\partial x}. \tag{2.1}$$

Deconvolution

Deconvolutions can be thought of as reversing the process done in a convolutional neural network [41]. Essentially attempting to recreate the input from the output activation by running the CNN in reverse top-down. The convolutions get replaced with deconvolutions also called transposed convolution. The filter values are copied after transposing their values. The process also replaces max-pooling layers with

unpooling operations wherein the feature map is upsampled depending on the pooling parameters while retaining the maximum value. This is done by storing the position of the maximum value in the forward operation of the CNN. The process is repeated from the layer whose filter is to be visualized back to the input space.

Guided Backpropagation

Guided backpropagation [42] builds upon deconvolution. It combines vanilla backpropagation at ReLUs (knowing which elements are positive in the previous feature map) with DeconvNets (keeping only positive gradients).

Input $\times$ Gradient

Input $\times$ Gradient is another type of attribution method wherein, the gradient was multiplied with the input x [94]. The equation to compute the Input $\times$ Gradient is

$$Input \times Gradient = \frac{\partial S}{\partial x}.x. \tag{2.2}$$

GradCAM

GradCAM is a generalization for Class Activation Map (CAM) as CAM limits the CNN to require a global average pooling layer at the end of the convolutional blocks [36]. GradCAM on the other hand does not require this.

For the k^{th} feature map activation A_k in the final convolutional layer of a CNN, the gradient of the score y_c for the class c of interest is initially computed. The average score of the gradient w.r.t. each node in the feature map is computed to get an importance value $\alpha_{k,c}$ for the particular feature map. The equation to estimate

$\alpha_{k,c}$ is

$$\alpha_{k,c} = \frac{1}{m.n} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{\partial y_c}{\partial A_{k,i,j}}. \tag{2.3}$$

Here, $A_{k,i,j}$ is a single neuron/node at position (i, j) in the feature map A_k of dimension m x n. GradCAM then linearly combines the importance score for each of the feature map and pass them through a ReLU the total relevance score map equals to

$$GradCAM = \text{ReLU}(\sum_{k}^{K} \alpha_{k,c} A_k). \tag{2.4}$$

The relevancy score is then upsampled using bi-linear interpolation to the same dimension as the input.

GradCAM++

GradCAM++ can be considered as a generalized formulation for GradCAM [37]. This method uses the second and third order derivative on the gradients to obtain the gradient weights α_{ij}^{kc} for the particular class c and the feature activation map k yielding the value of α_{ij}^{kc} as

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{{}_{ij}^{k\,2}}}{2.\frac{\partial^2 Y^c}{\partial {A_{ij}^k}^2} + \sum_a \sum_b A_{ab}^k (\frac{\partial^3 Y^c}{\partial {A_{ij}^k}^3})}. \tag{2.5}$$

Using these, the weights of GradCAM++ comes out to be

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc}.\text{ReLU}(\frac{\partial Y^c}{\partial A_{ij}^c}). \tag{2.6}$$

Multiplying these weights with the activation and passing the pooled value from all the filters through a ReLU gives the final GradCAM ++ heatmap representation for each of the inputs as

$$GradCAM++ = \text{ReLU}(\sum_k w_k^c.A_{ij}^k). \tag{2.7}$$

Guided GradCAM

Guided GradCAM is a combination of GradCAM and Guided Backpropagation to obtain pixel-level granular GradCAM representation [39]. GradCAM is combined with Guided Backpropagation by performing an element-wise product of the two to obtain Guided GradCAM.

Layer Wise Relevance Propagation

Layer wise Relevance Propagation (LRP) redistributes the prediction score for a particular class of interest through custom backward pass through the model back to the input following a conservation principle [34].

DeepLift

DeepLift is similar to LRP in the sense that it decomposes the output prediction for a particular input by backpropagating the contribution of all neurons in the model to each feature of the input [35]. DeepLift gives a measure of the change in output from a "reference" output w.r.t. the change in input from a 'reference' input. The reference is a neural input that is task-irrelevant. Here an array of zeros is used with the same dimension as the input [65].

ScoreCAM

ScoreCAM is a perturbation-based expansion to the class activation map framework [40]. ScoreCAM basically tries to mask part of the input and observe the change in prediction score for the class of interest similar to the occlusion approach. However, unlike occlusion, here the mask is obtained by initially forward passing to get the feature map activation. To perturb the input these are up-sampled to input dimension and smoothed by normalizing to have a value between 0 and 1. Later they

are masked based on the activation scores and the masked input is fed into the CNN to compute prediction score which serves as a weight for the feature map. This process is repeated for all the filters present in the final convolutional layer and pooled to obtain the final ScoreCAM representation.

FullGrad

FullGrad is an attribution method that aggregates the gradient for the entire network by decomposing the prediction score into input sensitivity and per neuron sensitivity components. FullGrad computes the gradient of the biases from the entire network and sums them [33].

LayerCAM

LayerCAM builds on top of GradCAM wherein the class activation maps are extracted for all layers instead of the final convolutional layer as is done in CAM/GradCAM [38].

2.2.5 Metrics

The visualization approach assigns a relevancy or importance scores to each pixel/ data point in the input. To compare different explanation methods, metrics to quantify the similarity the explanations are after randomization as well as, the efficiency in capturing the true underlying ground truth is equally important. For the robustness measure, both the Pearson's correlation and Structural Similarity index (SSIM) [95] were used to compare explanations before and after randomization. The output of the visualization method's being compared here can be considered as images with relevancy scores on a pixel basis. SSIM has been demonstrated to have good

agreement with human observers when using reference images by quantifying the perceptual difference and have been shown to perform better compared to both mean squared error as well as the peak signal-to-noise ratio [96]. In addition, the correlation coefficient further quantifies the linear relationship between the two. Ideally, for a robust method, the original explanations should become uncorrelated or minimally correlated w.r.t. explanation after randomizing.

Robustness Metrics

The measures used to compare the similarities between the explanations are adapted from Adebayo et al. [78].

1. Pearson's Correlation Coefficient: Compute the sample correlation between the explanations yielding a measure of the strength and direction of the linear relationship between the two variables. Here the explanations would initially be flattened out. The equation to compute the Pearson's Correlation Coefficient is

$$r = \frac{cov(x, y)}{\sqrt{var(x)}.\sqrt{var(y)}}. \tag{2.8}$$

2. Structural Similarity Index (SSIM): Measure of the perceptual similarities between two images SSIM. Given two images/inputs SSIM provides a measure of distortion along the luminance, contrast, and correlation dimensions [97].

Sensitivity Metrics

To compare the effectiveness of these models in identifying the true signal of interest, two measures to quantify the sensitivity are used. The main goal of evaluating these measures is to ensure that a majority of the top relevancy scores assigned fall in the ground truth region of the data. Ground truth mask is a binary array with a value of one assigned to all non zero data points in the ground truth and a value

of zero for others. The relevance mass accuracy measure quantifies how much of the total relevancy assigned by the methods is localized in the ground truth region. This gives a measure of accuracy.

1. Relevance Mass Accuracy (RMA): Ratio of the total relevancy inside the ground truth mask divided by the sum of the total relevancy assigned for the input [98]. The equation to compute RMA is

$$RelevanceMassAccuracy = \frac{R_{within}}{R_{total}}. \tag{2.9}$$

   Since in the simulation the source signal has been assigned to a dipole that projects onto the surface, a non-zero ground-truth value is assigned to all channels due to volume conduction. Therefore, to compare the similarity with the ground truth topoplot representation, a different distance measure of similarity is used for spatial data

2. Cosine Similarity (For Spatial Sensitivity): Cosine similarity computes the cosine of the angle between two non-zero vectors which is equivalent to the inner product of the vectors after normalizing to get unit length [99]. The equation to compute cosine similarity is

$$CosineSimilarity = \frac{A.B}{\sqrt{\Sigma A}.\sqrt{\Sigma B}}. \tag{2.10}$$

2.3 Results

The cross-validated robustness and sensitivity measures were estimated for each of the three conditions for different levels of SNR. Each of the following subsections gives the comparison for each of the conditions.

2.3.1 Event Related Potential Component (Temporal Precision)

The averaged cross-validated performance metrics are summarized in Fig 2.8. From the RMA measure, Deeplift was found to be the most accurate/sensitive followed by LRP and I×G to localize the ERP component. This was followed by Guided GradCAM and LayerCAM. On the other hand, GradCAM++ was the worst at temporal precision, followed closely by GradCAM and ScoreCAM.

Looking at the top 5 percentile explanations, DeepLift still emerged as the best followed by LRP, I×G. However, Saliency, GradCAM, and LayerCAM become much more comparable in the top 5 percentile. Even here, GradCAM++ remained the worst.

However, when the similarity of original explanations were compared to that with randomized labels, it was observed that methods like GradCAM++, Fullgrad, and Saliency have very similar explanations suggesting that their explanations are not class-specific. Similarly Deconvolution and Guided Backpropagation also yielded high correlation with the original true explanation. DeepLift, LRP, I×G, and GradCam were the most robust.

In the case of randomized weights, Deconvolution, Guided Backpropagation had the highest R-value followed by GradCAM++. For SSIM, GradCAM++ had the highest value followed by Saliency, FullGrad and ScoreCAM. DeepLift, LRP, I×G were still having low values.

Overall Deeplift was found to be the best closely followed by LRP and I×G,. They had a good trade-off in both robustness and sensitivity whereas GradCAM++ was the worst. Even though Saliency, Guided Backpropagation, and LayerCAM had good sensitivity, they were not very robust to randomizing labels and weights.

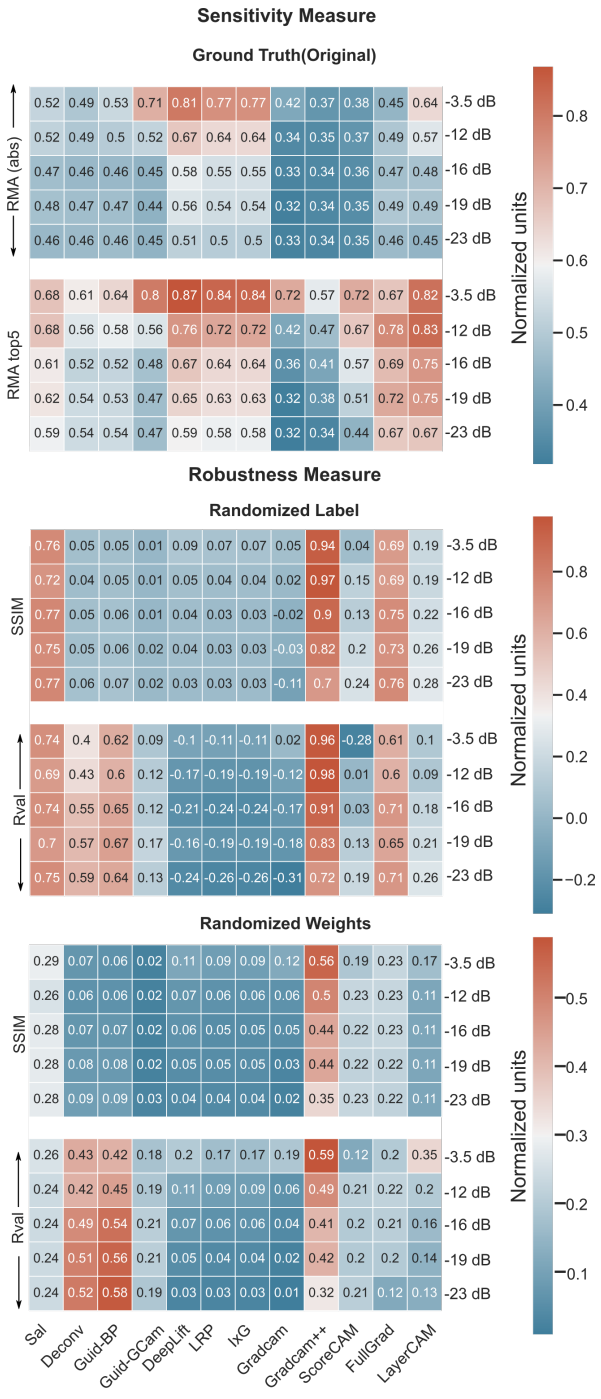

Figure 2.8: Comparison of the cross-validated metrics for different explanation methods with and without label/model weight randomization for detecting ERP components.

2.3.2 Spectral Perturbation (Frequency)

The averaged cross-validated performance metrics are summarized in 2.9. From the RMA measure, most measures do have high accuracy but Deeplift was still the most accurate/sensitive method. This was closely followed by LRP, I×G, Guided G-cam, Guided Backpropagation, Saliency, Deconvolution, and FullGrad. ScoreCam, GradCAM++ was the worst followed by GradCAM and LayerCAM.

However, when the similarity of original explanations to that with randomized labels is compared, like before it was observed that GradCAM++, Fullgrad, Saliency, Guided Backpropagation and Deconvolution have very similar explanations suggesting their explanations are not class-specific. Similarly LayerCAM and ScoreCAM also yielded a high correlation with the original true explanation. DeepLift, LRP, I×G, GradCAM, Guided GradCAM were the most robust.

In the case of randomized weights, GradCAM++, Deconvolution, Guided Backpropagation had the highest R-value followed by Saliency, FullGrad, ScoreCam, LayerCAM, and Guided GradCAM. For SSIM, GradCAM++ had the highest value follower by Saliency, FullGrad, and ScoreCam. DeepLift, LRP, I×G, GradCAM were still having low values.

Overall Deeplift was found to be the best closely followed by LRP and I×G. They had a good tradeoff in both robustness and sensitivity whereas GradCAM++ was the worst. Even though Saliency, Deconvolution, Guided BP, Guided GradCAM, FullGrad had good sensitivity they were not very robust to randomizing labels and weights.

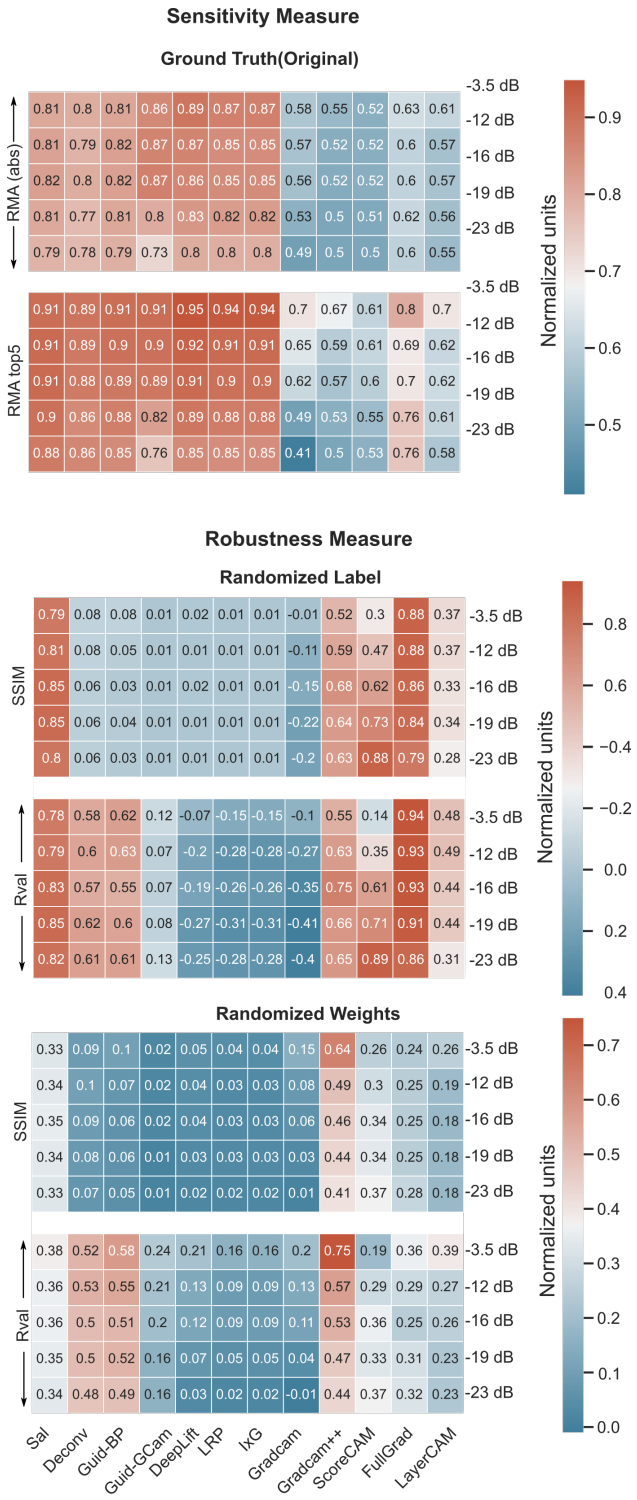

Figure 2.9: Comparison of the cross-validated metrics for different explanation methods with and without label/model weight randomization for detecting spectral perturbation features.

2.3.3 Scalp Distribution (Spatial)

The averaged cross-validated performance metrics are summarized in Fig 2.10. Here, cosine similarity was used instead of RMA as there exists non-zero ground-truth value in all channels due to volume conduction. Here, unlike other measures, based on cosine similarity, it was found that on the true explanation, GradCAM, ScoreCAM had the highest RMA followed by GradCAM++, FullGrad and LayerCAM. DeepLift, LRP, and I×G still had high values but were lower than the other measures. However, looking at the top 5 percentile, it has been found that LayerCAM had the highest accuracy followed by DeepLift and then LRP, I×G, and GradCAM. Guided Backpropagation, Deconvolution, and Guided GradCAM were the worst for spatial relevancy. Even though GradCAM has high sensitivity, their performance drops much fast with SNR lower than 19dB compared to other methods.

However, when the similarity of original explanations was compared to that with randomized labels, the measures like GradCAM ++, ScoreCAM, Fullgrad which had the highest sensitivity to ground truth, also had the most similarity to the randomized label explanation. Saliency and Guided Backpropagation also had high similarities to the original explanation. DeepLift, LRP, I×G, GradCAM, and Guided GradCAM were the most robust.

Similarly, in the case of randomized weights,GradCAM ++, ScoreCAM, Fullgrad which had the highest sensitivity to ground truth, also had the most similarity to the randomized label explanation. Saliency and Guided Backpropagation also had high similarities to the original explanation. In addition, randomizing weights had high similarity for LayerCAM as well. DeepLift, LRP, I×G, GradCAM, and Guided GradCAM still remain the most robust.

Overall GradCAM, Deeplift, LRP, and I×G were the better approach and had a good tradeoff in both robustness and sensitivity. GradCAM++ was the worst. Even

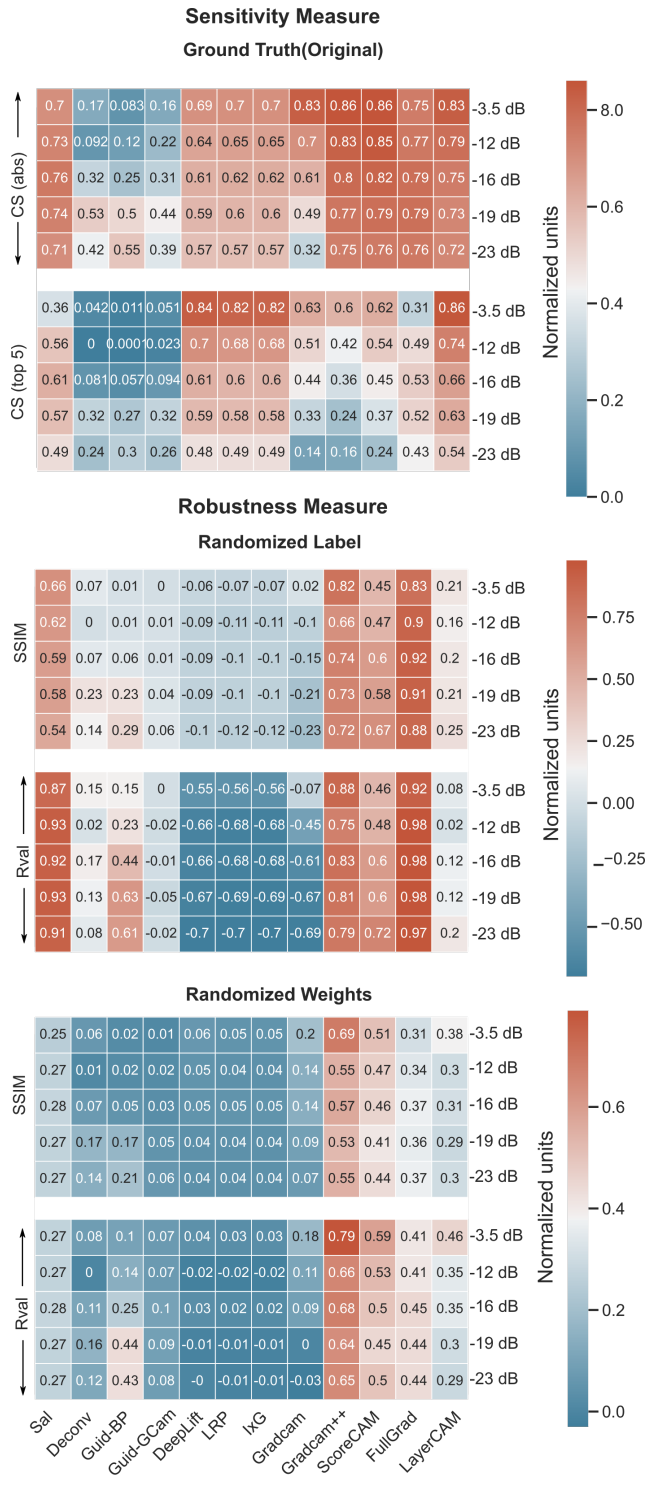

Figure 2.10: Comparison of the cross validated metrics for different explanation methods with and without label/model weight randomization for detecting spatial features.

though ScoreCAM, FullGrad, LayerCAM, and Saliency had good sensitivity, they were not robust to randomizing labels and weights.

2.4 Discussion

Including explainability approaches in deep learning studies is critical to understand the operation of the model, identifying the most relevant features with discriminative power, and generating scientific insights about the datasets. However, choosing these approaches require a good understanding of the strengths and weaknesses of the methods available when applied to EEG. Twelve heatmap-based visualization methods were systematically compared for their ability to detect different fundamental attributes of EEG. Using a simulation framework allows us to limit and understand the exact feature from which the model can learn from. Using real EEG, it is very difficult and challenging to ensure the model is only learning from a particular feature, and to know the true ground truth available, their location, duration, etc. For the same reason, it would be very difficult to compare the methods on how well they capture the ground truth signal as well. The robustness and the accuracy of these models to temporal, spectral, and spatial sensitivity of these methods for varying signal-to-noise ratios were compared. Figure 2.11 gives a high level summary of the different comparison. The methods which have a mean sensitivity measure greater than 0.55 is indicated by the dark blue color. Red color indicates the particular method for the condition being considered is not class specific (robustness measure > 0.5). Similarly, orange color indicates the method is not class specific with robustness measure > 0.3 but < 0.5. If the method is not model specific it is indicated by the asterisk "*" symbol. Here, if the robustness measure >0.5, they are marked with "**" and if the robustness measure >0.3 and < 0.5, it will be indicated by a single "*".

Based on these comparisons, some of the recommendations for different conditions

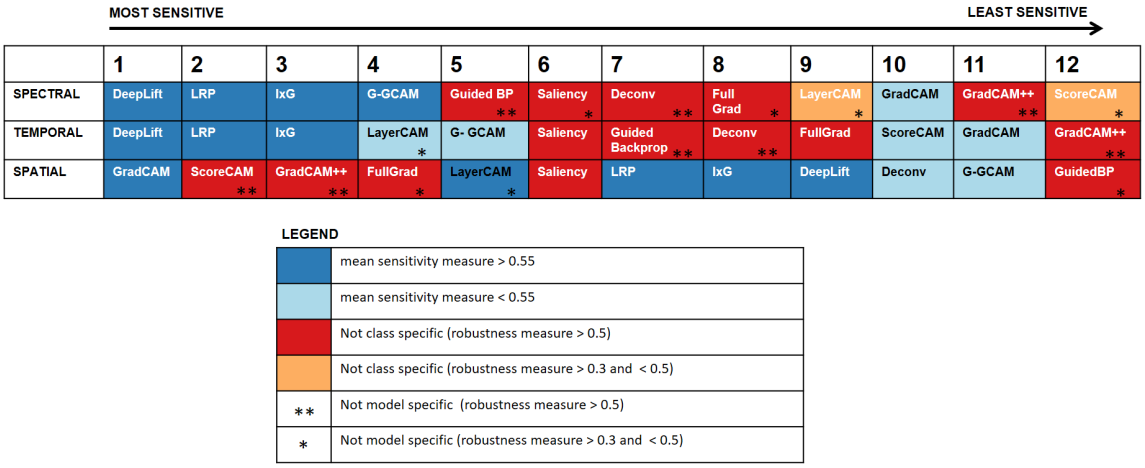

Figure 2.11: Comparison of the cross validated metrics for different explanation methods with and without label/model weight randomization for detecting spatial features.

are summarized in table 2.3.

Evaluating the robustness and sensitivity measures, even though many measures show high accuracy/sensitivity to the feature of interest, they are not class or model-specific. E.g., Saliency/Gradient is a basic yet one of the most commonly used model explanation methods in EEG [59],[60],[57],[61]. They also have high sensitivity to detect spectral perturbation and relevant channels as well. However, randomizing the model weights or labels yielded a very similar explanation to the original one. This suggests that they are not model or label-specific. Therefore, this method should be used with caution. A similar observation was found for many of the methods like Deconvolution, Guided Backpropagation, ScoreCAM, FullGrad, LayerCAM, GradCAM ++ as well. GradCAM ++ was one of the least reliable explanation methods.

On the other hand, DeepLift, Input $\times$ Gradient, and LRP was found to be both accurate as well as robust in all three cases (spatial, temporal, and spectral). Looking at the explanation metrics, LRP with epsilon rule, and Input $\times$ Gradient share very significant similarities. This is because previous studies have shown that when all the non-linearities involved are ReLU, epison rule-based LRP approximates to Input $\times$

Table 2.3: Recommendations for the use of different explainability approaches for EEG. The methods are arranged alphabetically in each column.

No.	**Temporal**	**Spectral**	**Spatial**
Sensitive	DeepLift I×G LRP	DeepLift Guided GradCAM I×G LRP	DeepLift FullGrad GradCAM GradCAM++ I×G LayerCAM LRP Saliency ScoreCAM
Robust	DeepLift GradCAM Guided GradCAM I×G LRP ScoreCAM	DeepLift GradCAM Guided GradCAM I×G LRP	Deconvolution DeepLift GradCAM Guided GradCAM I×G LRP
Methods to avoid	Deconvolution FullGrad GradCAM GradCAM++ Guided BP Saliency ScoreCAM	Deconvolution FullGrad GradCAM GradCAM++ Guided BP LayerCAM Saliency ScoreCAM	Deconvolution FullGrad GradCAM++ Guided BP Guided GradCAM LayerCAM Saliency ScoreCAM
Recommended	**DeepLift**	**DeepLift**	**DeepLift**
Alternatives	LRP/ I×G	LRP / I×G	GradCAM LRP / I×G

Gradients [100]. There exist multiple studies in Computer Vision that assessed the unreliability of Saliency map-based approaches [101] [78]. However, these studies do not measure the accuracy of these explanation methods. This is an important question because, in the study by Adebayo et al. [78], they identified that GradCAM was one of the most reliable/robust explanation methods available. In this study, we do show that even though the robustness aspect is preserved in all the 3 conditions, GradCAM is not ideal in the case of spectral perturbation and temporal data conditions. The reasoning for that comes from the framework itself. GradCAM as well as the general class activation maps, compute the model explanation w.r.t. the last convolutional block. With successive pooling and convolution operations, the temporal resolution of the activation in the final convolutional layer would be small. These methods get an estimate of the relevant input by performing a bilinear interpolation to upsample to the input dimension. These will lead to reduced temporal resolution, a key attribute in EEG. However, when we are not interested in the temporal aspect, but instead want to look at spatial relevancy, GradCAM was found to be the most accurate method. Another limitation of using GradCAM which needs to be checked for was that their performance decreased much faster than other methods when the SNR decreased i.e. when the model confidence dropped. One additional point to keep in mind if researchers plan on using GradCAM is that many of the existing EEG architecture uses a spatial convolutional layer in the initial layers. This spatially mixes the information across channels and the succeeding layers do not have channel-independent data. Therefore, using GradCAM in such a case will not be able to produce channel relevancy as the last convolution is purely temporal data. So, this study recommends researchers adopting heatmap-based model explanation methods to either use DeepLift or Layerwise Relevance Propogation in general to explain deep learning studies. However, unless the decoding is poor, GradCAM is still a good alternative for estimating spatial relevancy.

Overall, this study provides both a framework as well as an empirical comparison

of different model explanation methods. The sensitivity to detect three fundamental properties in EEG, specifically the temporal, spectral, and spatial properties were evaluated. Pitfalls in using some of these methods were identified. It was also observed that some methods were consistently better in all three aspects. Overall LRP or DeepLift was the most reliable method among all. They were also the most accurate in identifying the ground truth. Even though GradCAM is one of the most robust methods, they fail when the SNR is either low or in the case wherein temporal precision is critical.

2.5 Conclusion and future directions

The approach used here will serve as a benchmark for future researchers to get familiarized with the robustness and effectiveness of multiple explainable techniques; specifically different heatmap based attribution methods. The research provides a summary and recommendations to understand when some of these methods fail and what they can capture in EEG. This study is limited to features that are commonly reported in the tasks studied in this research. There could be many other features to test for and the set is not exhaustive. Overall, this research identified that some of the most used model explanation methods such as Saliency/Gradient are not class or model-specific. It was found that DeepLift was consistently accurate as well as robust to detect the three key attributes tested here. GradCAM even though was consistently robust, does not have good temporal precision. However, it is still good for detecting spatial patterns for signals with high SNR. The next chapter will demonstrate how these methods can be used to debug the model when applied to real EEG as well as show how they can capture the underlying brain dynamics. The method when added to existing studies will provide additional context to evaluate the bias of the models to spurious correlations or artifacts. Some of the limitations and future directions of the analysis are discussed below:

2.5.1 Approximation error

Synthetic EEG is only an approximation to measured EEG. Many physiological and non-physiological signals and artifacts, which are generally present in measured EEG, are not contained in the synthetic EEG. This can be both an advantage and also a limitation of the simulation approach. There is a possibility of missing some key EEG properties while modeling using simulation. However, in this study, the objectivity was prioritized higher to compare the different methods. Moreover, EEG data is quasi-stationary, context-dependent, and influenced by learning. Thus, interpretability models must also account for these factors if they are part of the experimental design. In future studies, with the developed framework, identified confounds and complex modeling could be added later.

2.5.2 High level explanations

The scope of this research is limited to visualization methods that highlights key segments of the input data. But assessing which specific feature in EEG caused the correct prediction would still be difficult to ascertain. However, combining the methods can help develop insights. Knowing the scalp relevance heatmap can help isolate the relevant channels. Later, checking the relevancy of temporal data can get specificity for temporal localization. Following this with activation maximization [71] on these channels or other feature perturbation approaches [72] can give insight into the relevant frequency bands or feature that is being perturbed. This can be followed up with traditional signal processing methods focused on the relevant regions to gain additional insights. This method can identify which features are not sensitive (if any) as well as the regions that are not important and those can be avoided.

2.5.3 Other approaches

Although this research limited the analysis to visualization-based approaches, there are other types of model explanations as summarized in the introduction. Some of these methods could provide better insights. However, exploring all of this iterations is outside of the scope of the study and will be explored in future studies.

Chapter 3

Decoding Neural Activity Preceding Balance Loss during Standing with a Lower-limb Exoskeleton

3.1 Introduction

The World Health Organization (WHO) reported that over 37 million falls require medical attention each year worldwide [102]. Indeed, falls are a leading cause of injury, loss of independence, hospital admission, and even death. While conventional therapies have been successful in fall reduction and prevention, many individuals with severe illness or injury remain unable to participate in activities of daily living (ADLs) or complete standard care protocols. Recent efforts to aid these populations have utilized wearable robotic systems and, in particular, powered robotic orthoses (i.e., exoskeletons) [103],[104].

The U.S. Food and Drug Administration (FDA) classifies powered exoskeletons as Class 2 medical devices with special controls. They are used frequently for rehabilitation applications due to their ability to provide active, assistive support for walking, sitting, and standing [105],[106]. When compared to traditional therapies, these devices provide intense training in an active and stimulating environment while providing quantifiable markers of progression [107],[108]. In addition to rehabilitation, exoskeletons can also be purposed to reduce the risk of falling and/or aid in fall prevention.

However, falls while wearing the exoskeletons are a significant risk in using these devices [109]. Current FDA-cleared exoskeletons use different strategies for dealing with potential falls and are indicated for use with a trained companion. The effectiveness of these strategies is not studied and is still unclear. Some systems utilize kinematic response assessments to detect fall events based on accelerometers, magnetometers, or joint angles. The Indego and Ekso exoskeleton systems detect falls in real-time by checking for excursions in kinematic variability beyond certain limits. In the case of the Indego device, movements beyond a set threshold will trigger corrective postural movements to reduce the risk of injury [109]. However, while other studies have examined fall risk and incidence [110], tested exoskeletons during perturbations [111],[112], or even developed positioning algorithms to promote safer falls [113], very few appear to both detect and respond to these falls or perturbations. There was only one study that was identified which detailed an exoskeleton system with built-in perturbation or fall detection and response. In this study, Monaco et al. utilized a micro-controller to compare real-time kinematics with predicted walking values. Threshold reaching discrepancies between the predictions and real values were used to apply corrective hip torques to restore balance. Their detection algorithm was able to identify the lack of balance resulting from slippages within about 350 ms of the event [114]. Nevertheless, there are still drawbacks to this mechanism of fall detection; kinematic measures leave minimal time between detection and the fall event. In these systems, given that the use of electric motors with large gear reductions will have reduced response speed, early detection of balance loss is critical. With this in mind, approaches that can identify and act to correct balance loss earlier would be extremely beneficial.

Kinematic measures are not the only way of detecting fall events. Multi-sensory information from visual, somatosensory, and vestibular systems acting on the cerebral cortex, cerebellum, and brainstem have a significant role in postural corrections [115]. These sensory signals might precede the latency of kinematic responses and could

offer a longer stimulus to fall interval within which to respond. Physical balance perturbations elicit cortical responses called Perturbation Evoked Potentials (PEP). These PEP can be detected using electroencephalography (EEG). A PEP generally consists of 3 components. The first component is a small positive wave (P1) at approximately 30-90 ms. This is followed by a negative peak at around 90-160 ms with a final, late response (P2 and N2) around 200-400 ms [88]. These PEPs are typically observed by averaging waveforms across many trials. However, if PEP could be detected from a single trial, balance perturbations could be identified much earlier. This would afford considerable, additional time to initiate preventative movements.

Studies examining perturbations during exoskeleton use with an EEG paradigm, as well as the temporal relationship between signal modalities, are rare [116]. More importantly, to our knowledge, no previous studies have evaluated the influence of balance perturbations on EEG during exoskeletal suit use. Further understanding of the influence of exoskeletons on physiological responses observed with EEG as well as physical responses to perturbations is important. In this study, how different perturbations during standing conditions modulated the brain activity was evaluated and tested the possibility of detecting physical perturbations from single-trial EEG in individuals wearing an exoskeleton.

3.2 Methods

3.2.1 Participants

Seven healthy participants (5 male) aged 18-32 participated in the study. The experimental protocol was approved by the Institutional Review Board (IRB) at the University of Houston, in accordance with the Declaration of Helsinki. The written informed consent form was collected from each of the participants before the start of

the experiment.

3.2.2 Experimental Setup

Participants were fitted with a 64-channel EEG cap (ActiCap, Brain Products, GmbH, Morrisville, NC) referenced to the ear lobes. 60 active AG/AgCL electrodes were placed in the cap according to the modified 10-20 international system to record EEG signals. Electrodes normally positioned at FT9 and FT10 were moved to replace the AFz and FCz electrodes on the cap (ground and reference, respectively). In addition, electrodes that were to be placed at TP9, TP10, PO9, and PO10 were instead used to measure electro-oculography signals (EOG). Two electrodes were placed above and below the right eye with the remaining two electrodes placed at the lateral canthus of each eye to extract the eye-related artifacts. EEG/EOG data were recorded wirelessly using the MOVE system at 250 Hz and amplified using the BrainAmp DC amplifier (Brain Products, GmbH, Morrisville, NC).

Surface electromyography (EMG) sensors were placed over the tibialis anterior (TA), Medial Gastrocnemius (MG), Lateral Gastrocnemius (LG), and Soleus (S) muscles of both legs, along with one sensor on the forehead and torso. EMG data were collected wirelessly using the Delsys Trigno system (Delsys Inc., Boston, MA).

After set-up and electrode impedance measurements, participants were asked to stand comfortably on a balance platform (Neurocom Balance Manager platform, (NeuroCom, Clackamas, OR) for 2 minutes to acquire eyes open resting-state activity. At the end of 2 minutes, subjects received a series of postural perturbations. This consisted of a series of 32 constant (duration, period, and velocity) perturbations where the platform generated maximal backward translations (displacement of 6.35 cm in 400 ms, i.e. velocity of 15.875 cm/s). This condition is referred to as the Random Timing Condition (RTC), as the timing alone was randomized. The second postural task consisted of 33 random/unexpected perturbations where the platform

generated forward/backward/tilted perturbations in a random order (Random Timing and Type Condition - RTTC). Individual trials with the same parameters as the RTC trials were embedded randomly into the RTTC condition. After 16 trials of RTC and RTTC, respectively, a break of approximately 2-5 minutes was given to avoid fatigue. Each trial lasted five seconds and the timing to perturbation onset was randomized in all trials to avoid anticipation of when the perturbation would occur. All conditions were repeated with and without the H2 exoskeleton (in passive mode with the joints decoupled) to evaluate if PEPs would be altered in the presence of the mechanical constraints introduced by wearing the exoskeleton. For every other participant, the order of trials with and without H2 was reversed. The protocol is summarized in Fig 3.1.

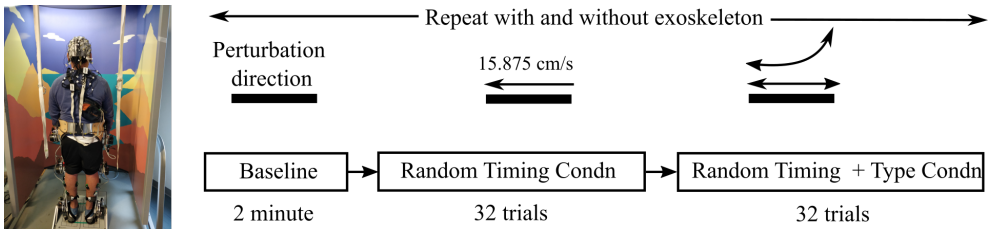

Figure 3.1: Experimental protocol: the two conditions were repeated with and without the exoskeleton. A 2-5 minute break was provided in between each of the blocks (RTC RTTC).

3.2.3 Signal pre-processing

The pre-processing steps used to process EEG, EMG, and the Neurocom data are summarized in figure 3.2.

Both the EEG and EOG signals were bandpass filtered between 0.2 to 50 Hz to remove low-frequency drift and minimize muscular artifacts. A 4th order zero-phase Butterworth filter was used to avoid phase distortion. The high pass cut-off of 0.2 Hz was selected from Tanner et al., which suggested high pass filtering above 0.3 Hz will distort the ERP components [117]. Ocular artifacts were removed using the

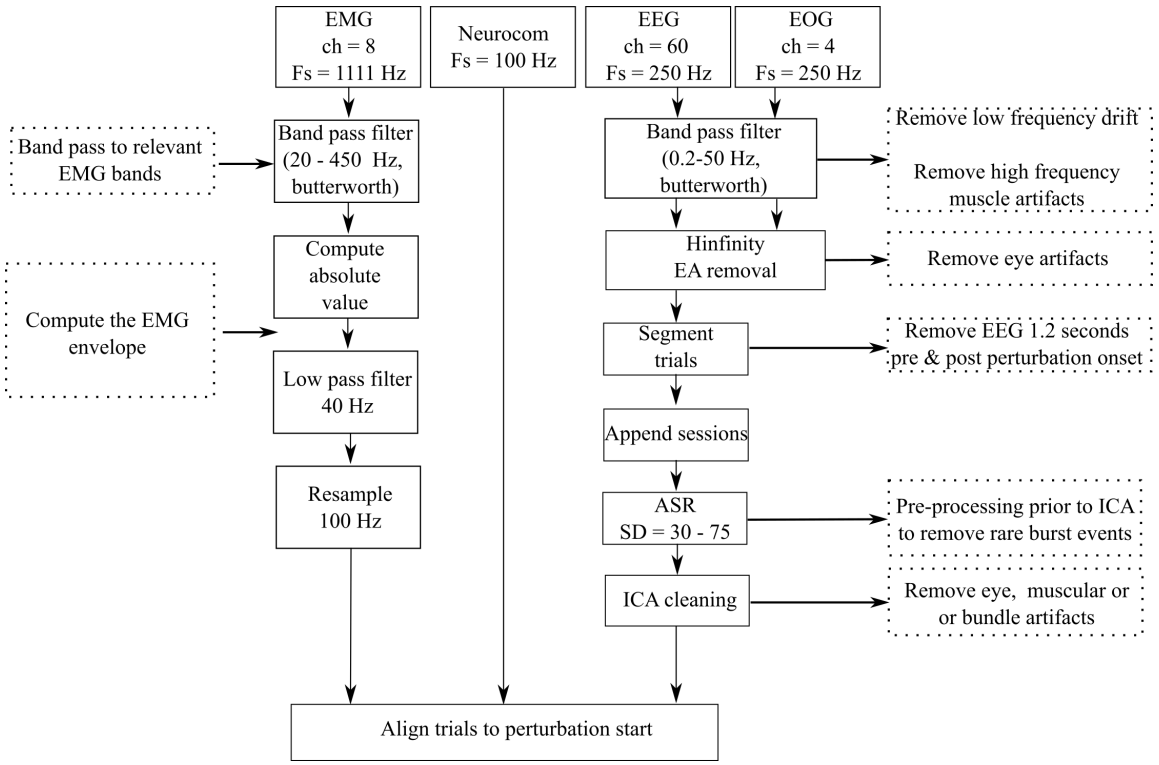

Figure 3.2: Flowchart detailing the different pre-processing steps performed for each of the signal modalities.

H-infinity-based adaptive filter [118]. The *gamma* parameter was set to 1.1 and the q parameter used was 1e-11 from empirical testing. Data 1.2 seconds before and after the perturbations were discarded and individual trials were concatenated together. Later, to remove any sudden spikes in the EEG and improve Independent Component Analysis (ICA) decomposition, Artifact Subspace Reconstruction (ASR) [119] with less conservative thresholds of 30-75 were used to reconstruct poor components in artifactual windows. The thresholds were selected based on empirical evaluation and also by recommendations from Chang et al. [120]. ICA decomposition was then performed using the Infomax algorithm to identify and remove ocular, muscular, or bundle artifacts (artifacts caused by the physical pulling of cable bundles). Here, a more conservative cleaning is performed to remove 26-44 ICs across subjects. Ocular artifacts were identified by looking at topographical distributions, power spectra with power localized in the delta/theta bands, as well as the time-series data for repeatable

ocular artifacts. Muscular ICs were identified by examining the spatial weighting of the IC (localized in the temporal channels), power spectra (looking at the increasing power in 30+ Hz) as well as time-series data for spiking activity. The bundle artifacts were identified by the spatial weight of the IC (alternating pattern for the 2 bundles). Any additional ICs (indicating electrode shifts) were identified and removed. Representative examples of the ICs removed are provided in the supplementary materials. All the pre-processing steps were implemented using the EEGLAB library[121].

EMG data were bandpass filtered with a passband frequency of 20 - 450 Hz using a 4th order Butterworth zero-phase filter. Later, to extract the envelope, data were rectified by computing the absolute value and passing through a second low pass filter at 40 Hz. The envelope of the EMG was then resampled to 100 Hz to match with the sampling rate of the kinematic data from the Neurocom. All three modalities were then aligned to the perturbation onset in each of the trials.

3.2.4 Latency relationship between the signals

To study how electrophysiological and kinematic responses varied in response to the perturbation, all the signals after baseline correction were trial averaged. This also increased the signal-to-noise ratio. Averaging was done separately for each of the conditions. The period between -500 ms to -200 ms was used to estimate baseline correction values. Perturbation response in the first trial was consistently, significantly larger than the succeeding trials, and thus were removed before averaging. The trial averaged physiological and kinematic signals were aligned to the perturbation onset to evaluate the latency difference between the signals. In the end, the grand average response was computed by averaging the time series across all subjects and trials.

3.2.5 Detecting perturbations from single trials

A CNN was implemented to detect the presence of perturbations from 200 ms long windows of single-trial EEG. Class 1 was composed of individual trial windows during the baseline period (1200 to 500 ms) prior to the onset of perturbation. Class 2 consisted of EEG segments between -200 ms until +500 ms post perturbation onset. Windows of 200 ms from each of these classes were extracted in a sliding window manner with a one-sample difference. The data were scaled by dividing by a value of 100 (μV). The baseline period per trial was selected as class 1, instead of the resting state, to avoid the model prediction being confounded by impedance change between the two segments. It further ensures that internal states unrelated to the perturbations are comparable across the classes.

To increase the sample size for the classifier, trials not involving the exoskeleton were also included. Therefore, a total of 60 trials of RTC trials were used for training the model. Trials were randomized and divided into train, test, and validation sets. 15% of trials were divided into validation and 15% into the test set. The data was divided based on trials and not by random sampling of all the windows. This was done to avoid any potential data leakage due to the high level of overlap. This ensured that there was no shared information between the three sets. A total of 5 such held out sets were created for cross-validation to evaluate the generalizability of the model. In addition to test accuracy, F-score was also computed for each of the folds.

The architecture for the model is summarized in Fig. 3.3. The input to the model is the 200 ms EEG window (batch size x 50 samples x 60 channels). The model consisted of 5 temporal convolution layers of 8 units each (3 x 1 kernel size with a stride length of 1). A temporal pooling layer of 2x1 pooling dimension with a stride length of 2 was also used after every pair of convolutional filter layers except the last block. These should help with the trial-by-trial translational variance of the PEP components. The output from these convolutional layers was flattened and fed into

a dense, fully connected layer of 16 hidden units followed by an output layer with softmax activation.

A dropout layer with alpha = 0.5 was added in between the dense layer and the output layer to reduce overfitting. Except for the output layer, the model utilized ReLU as the activation function. An Adam optimizer [122] with a learning rate of 0.0001 was used to train the model. A batch size of 32 and epoch length of 100 was set. An early stopping condition was set to avoid the model from being overfitted. This stopped the training if the validation loss did not improve in 5 consecutive epochs. A re-initialized independent copy of the same model architecture was used for each fold and subject. The proposed model was implemented in python 3.6 using keras 2.15 [123] wrapper using Tensorflow [124] backend.

The model architecture was selected to better facilitate the GradCAM algorithm in identifying relevant channels. Most currently available models use a spatial filter in the early stage of the architecture. If spatial filters are used early on, the deeper layers can only see a mixed channel (time x number of filters dimension) representation. GradCAM will not be able to identify the relevant channel distribution. Here, the emphasis was put on explaining the model decision to ensure the model is indeed learning from relevant components and not driven by irrelevant signals. To ensure that prioritizing explainability during architecture selection did not impair decoding performance, the performance of the model was compared with the DeepConvNet architecture [72]. The original paper that proposed the DeepConvNet architecture used a 2-second long EEG, sampled at 256 Hz as input. For the DeepConvNet, to account for the difference in dimensions the architecture hyper-parameters were modified to make it compatible with our data. In this study, three blocks were used instead of four as the window size is not long enough to accommodate the 4th block. Additionally, to evaluate the impact of denoising, the process was repeated by training the model used in this study on EEG data prior to ICA cleaning instead of

the denoised EEG.

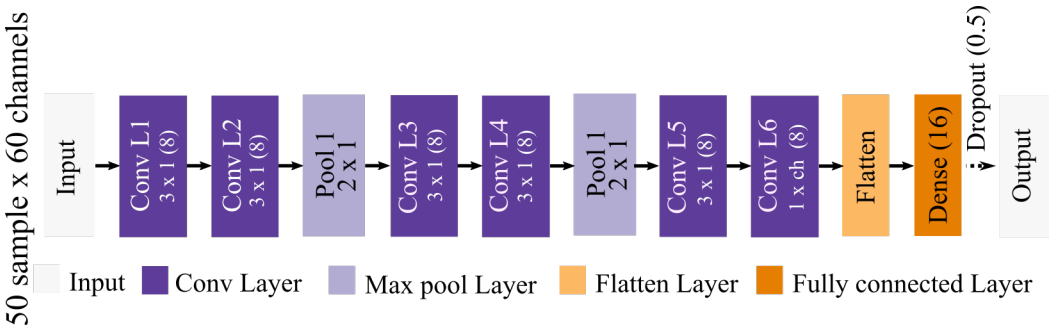

Figure 3.3: Model architecture: Each block correspond to different types of layers in the model. The dotted line is to illustrate the dropout operation during the training phase aimed at reducing overfit. During inference, all units were retained.

3.2.6 Explaining the CNN model decision

The model decision explanation was carried out using the GradCAM method [36]. GradCAM is a class-specific explanation technique that identifies relevant regions in the input that the model used to make the prediction pertaining to a specific class. The algorithm is explained in Selvaraju et al. 2017 [36]. GradCAM is a generalization for Class Activation Map (CAM) as CAM limits the CNN to require a global average pooling layer at the end of the convolutional blocks. GradCAM on the other hand does not require this. GradCAM computes the gradient of the score of the class of interest with respect to each of the feature map activations of the penultimate layer being considered. These gradients are then global average pooled to serve as weights for the particular feature map. A weighted sum of the feature map activations with respect to these weights is then computed. These are then are passed through a ReLU operation to consider only positive values as they contribute to making the correct prediction. Here, the penultimate layer used is the convolutional layer L5 to learn channel relevancy. From the model explanations, time-averaged GradCAM is computed to identify the relevant channels per window.

Next, k-means clustering was performed on the model decisions. All the correctly predicted data points across all subjects from the best performing fold (combined validation and test set) were fed into the clustering algorithm instead of visualizing hand-selected examples to avoid bias. The distance measure used was squared euclidean with the maximum number of iterations allowed set to the total number of samples present. The optimal cluster number was selected using the elbow method. K-means was evaluated for a variable number of clusters ranging from one to 100. The total within-cluster sums of point-to-centroid distances were computed for each of the K values. The K values that corresponded to the knee of the curve were selected. Instead of manually selecting the knee point which could be subjective, the Kneedle algorithm was used to detect the knee [125]. The parameter S was set to 0 as recommended in the offline setting in the original paper[125]. The process was repeated 5 times and the average K values were chosen for the final k-means clustering. The cluster results were then evaluated to assess whether the model was learning from the PEP components and not being driven by artifacts. The process was repeated on separate models trained on pre-processed EEG as well as raw EEG without ICA cleaning.

Post-hoc test to evaluate model explanation with traditional signal processing approaches

To evaluate how the network dynamics evolve with time during the PEP, a measure of dynamic functional connectivity called phase difference derivative (PDD) [126] was calculated for each trial. PDD is a measure of the stability of phase difference between two signals. It computes the instantaneous phase of the signal based on the analytic signal extracted from the Hilbert transform of each of the signals. For phase-locked signals, the difference in phase remains constant across time, in which case the derivative of that should be approximately zero. Taking the negative exponent of

the derivative further ensures that it is bounded between 0 and 1 with a value of 0 meaning no coupling between the signals. The equation to estimate PDD is

$$PDD_{ij}(t) = \exp(-|\frac{d\Delta\Phi_{ij}(t)}{dt}|). \tag{3.1}$$

Here, $\Delta\Phi_{ij}$ is the phase difference between signals i and j at time t. The PDD in the alpha band was calculated by initially band-pass filtering the signal using a 4th order zero-phase butter worth filter in the band (8-13 Hz). The PDD was estimated with a center frequency of 10 Hz and a window size of 128 ms. The window size was selected such that it contains at least one cycle of the lowest frequency of interest (8 Hz). The measure was estimated from seven channels. Six of the channels were relevant to the task (based on model explanations from CNN). A seventh channel, which we expected to be task-independent (TP7) was also evaluated. The PDD was baseline corrected (w.r.t. -500 ms to -200 ms) to further remove any residual connectivity across channels that are not task-dependent. The grand average ERP and PDD were estimated from each of these channels using the same procedure as described in the section above.

3.2.7 Continuous decoding of COP from EEG

The predictive power of EEG to continuously decode the COP variations in response to perturbations was then evaluated. Gated Recurrent Units (GRU) were used to decode the COP values. Considering the perturbations were solely a backward translation, only the y component of COP was decoded as it had the largest modulations. To evaluate the ideal model parameters, a hyperparameter search was performed by varying the number of layers (1 to 3) and the number of units per layer (8, 16, 32, 64, 128, 256, 512, 1024). This was followed by a dense layer with a ReLU activation function and the number of units equal to that of the GRU units. The dense layer was then connected to the output layer with a linear activation function.

To evaluate the decoding performance, the coefficient of determination (R2 score), Pearson's correlation coefficient (R-value) and mean squared error (MSE) metrics were used [127]. All of these were implemented in python using the Scikit library [128]. Similar to the classification model, 70% of the trials were divided into training, 15% for validation, and 15% for testing. The GRU model was trained and tested on five such splits to evaluate generalizability. Here, unlike the classification model, EEG from 1.2 seconds prior to perturbation onset until 1-second post perturbation onset was used. Separate models were trained for each combination of participant x number of layers x number of GRU units x folds. Predicted and actual COP values were evaluated using the measures on the validation set across all 5 sets to identify the optimal model hyperparameters. Upon identifying the optimal hyperparameters for the model with minimal computational cost, the optimized model was evaluated on the test set to determine final performance values. The models were trained using the Keras library with the TensorFlow backend. The initial learning rate was set to 0.001 with the model weights optimized using Adam optimizer [122]. The batch size used was 128 and trained for a maximum of 200 epochs with an early stopping condition of stopping the training if the validation loss did not improve in 5 consecutive epochs. The GRU was trained to minimize the mean squared error between the actual and predicted COP values. To further evaluate how the model generalized when the person was not only blind to the timing but also the type of perturbation, trials with the same type of perturbations that were randomly present in the RTTC sessions were also tested.

3.3 Results

3.3.1 Latency relationship between the signals

Fig 3.4 depicts the grand average response across channels during the exoskeleton RTC condition. The top row shows the grand average PEP components in the Cz EEG channel. All the previously reported components of the PEPs including P1, N1, and P2 are retained while wearing the exoskeleton. In addition, the P1 peak (75 $\pm$ 8 ms) and N1 peak (137 $\pm$ 12 ms) precedes the peak in EMG (MG: 195 $\pm$ 27 ms; LG: 182 $\pm$ 19 ms; TA: 180 $\pm$ 14 ms; S: 181 $\pm$ 13 ms) which again precedes the peak in the COP (365 $\pm$ 22 ms). The peak of COP is the point at which the participants start initiating the return to the original position. This indicates that EEG contains discriminatory information much earlier than the kinematic response which could be used to detect the balance perturbations.

3.3.2 Detection of balance perturbation using a convolution neural network

The capability for CNN to detect the PEP components and other underlying neural representations from single trials alone in a data-driven manner was tested. The cross-validated results are summarized in Table 3.1. Overall, all the subjects obtained above chance level ($\sim$ 50%) classification scores. A cross-validated mean test F score of 74.7 $\pm$ 4.5 % was obtained. Subject 4 had the lowest F score of 69.2 $\pm$ 7.1 % whereas subject 6 obtained the highest F score of 79.8 $\pm$ 1.9 %. The same model was tested on EEG without ICA denoising (Raw) and that model achieved a higher decoding accuracy (F score = 78.0 $\pm$ 5.2).

DeepConvNet yielded a mean test F score of 69.5 $\pm$ 4.3. Compared to DeepConvNet, our model performed better. However, we emphasize that the study do not

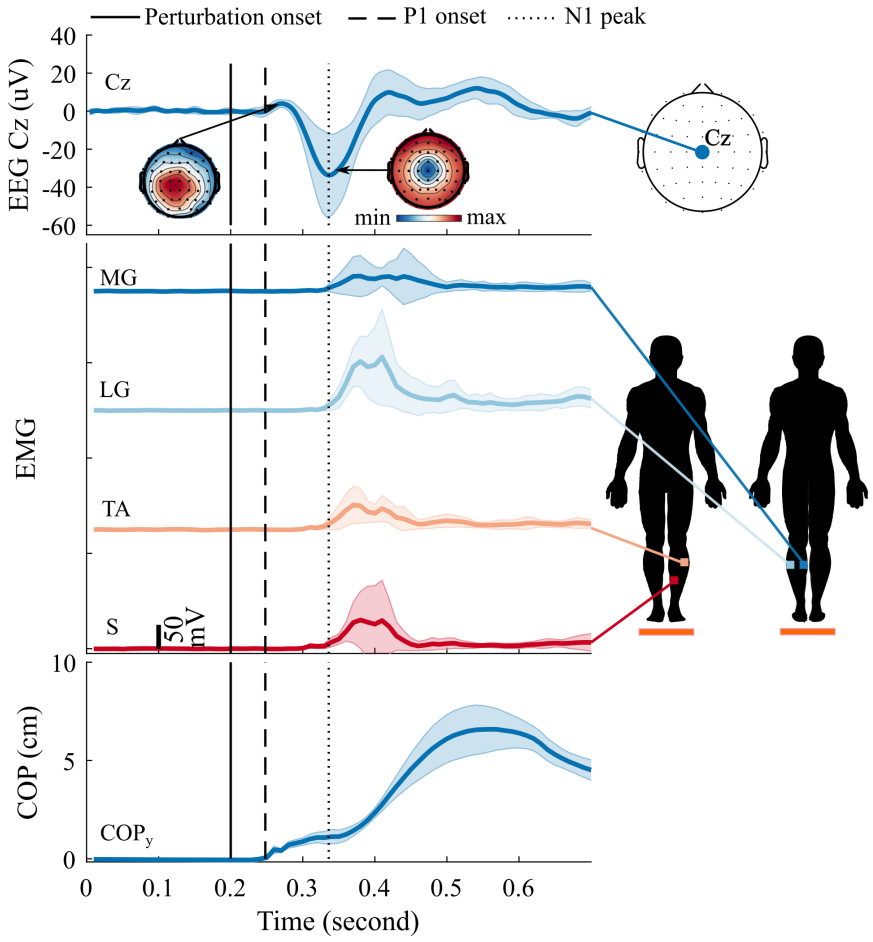

Figure 3.4: Between subject grand average latency difference between different electrophysiological and kinematic responses associated with balance perturbation while wearing the exoskeleton. The muscles shown are from the left leg with the following abbreviations: MG (medial gastrocnemius), LG (lateral gastrocnemius), TA (Tibialis Anterior), S (Soleus).

claim superiority for the architecture. Instead, this is evaluated only to show that focusing on architecture by prioritizing model explanation did not compromise model performance. To make the comparison fairer, randomization of the trials was made consistent for all models by assigning the same seed per fold.

Table 3.1: Cross validated performance metrics evaluated on the test set; all numbers are in percentages; Raw: model trained on EEG without ICA denoising, Clean: model trained on ICA cleaned EEG, DCN: DeepConvNet trained on ICA cleaned EEG.

Sub	**Accuracy**			**F-score**		
	Raw	**Clean**	**DCN**	**Raw**	**Clean**	**DCN**
S1	79.3 ± 4.1	75.2± 3.6	67.3 ± 2.5	79.2 ± 4.0	75.0 ± 3.4	65.4 ± 2.1
S2	72.2 ± 5.4	77.6 ± 6.8	73.3 ± 3.1	72.1 ± 5.4	77.5 ± 6.8	72.7 ± 3.7
S3	77.2 ± 11.5	75.5 ± 3.7	67.0 ± 5.5	77.0 ± 11.6	75.4 ± 3.7	65.3 ± 7.2
S4	84.1 ± 1.7	70.3 ± 5.4	65.0 ± 5.0	84.0 ± 1.6	69.2 ± 7.1	64.0 ± 4.9
S5	74.9 ± 4.2	71.4 ± 4.1	70.5 ± 5.2	74.5 ± 4.2	71.1 ± 4.1	69.9 ± 5.6
S6	81.7 ± 3.9	80.0 ± 1.8	79.8 ± 2.6	81.6 ± 3.9	79.8 ± 1.9	79.6 ± 2.6
S7	76.9 ± 5.3	75.3 ± 4.8	70.7 ± 4.1	76.6 ± 5.2	75.2 ± 4.7	69.7 ± 3.9
Avg	78.0 ± 5.2	75.0 ± 4.3	70.5 ± 4.0	77.9 ± 5.1	74.7 ± 4.5	69.5 ± 4.3

3.3.3 Explaining the CNN model decision

The optimal K value to perform the k-means on the model explanations was identified as 11 for the model trained on clean EEG and 14 for the model trained on EEG without ICA cleaning. Fig 3.5 shows the clustering results on the best-performing fold for both cases. Fig 3.5.a summarizes the clustering performed on the explanations from the model trained on cleaned denoised EEG. Fig 3.5.b corresponds to the explanations from the model trained on the raw EEG without ICA cleaning. The top row in both cases shows the mean relevancy score for the channels in each of the identified clusters. The middle row represents the distribution of window latency relative to perturbation onset (w.r.t. the last sample in each window). The distribution was normalized for visualization purposes. The third row shows the contribution of the examples in each cluster from each of the 7 participants.

From Fig 3.5.a, it can be seen that none of the clusters were weighing in on the periphery channels, which are often strongest if driven by artifacts. Almost all clusters were focusing on the channels in the motor, parietal and pre-motor regions to arrive at the decisions. From these, clusters C3 and C8 are localized in the Cz channel and are centered around the time when N1 peaks. Similarly, the parietal channels become

more relevant both in the early and late stages of the perturbations (C1, C6). The clusters localizing in the frontal channels (C7, C10) are centered in the latter half of the perturbation. In multiple clusters, the model is focusing on a broader range of channels but is still centered around the motor regions (C1, C2, C4, C11). The largest cluster, C5 had contributions from both the central as well as the parietal channels. Overall, in evaluating the spatial map distribution, the response is found to be highly dynamic, involving multiple brain regions varying over time.

To further verify that the model explanation was not biased against detecting artifacts and that the pre-processing was reliable and significant, the process of training and explaining the model decisions was repeated on EEG without ICA cleaning. The clustering results of data with artifacts are summarized in Fig 3.5.b. Even though the model trained on data without ICA cleaning achieved higher performance (F-score: 78 $\pm$ 5.2), evaluating the model explanations, it was observed that the model was learning the artifacts for decoding purposes. The model learned to detect the bundle artifacts indicated by alternative localized channel relevancy (C3, C6, C13) as well as started giving more relevance to the peripheral channels (C3, C5, C8, C10, C11, C12). These were absent in our pre-processed data.

Post-hoc test to evaluate model explanation with traditional signal processing approaches

The variability of the dynamic measure of the functional connectivity ΔPDD is shown in Fig 3.6. The parietal and parietal-occipital channels that are heavily reported to be involved with sensory processing have increased connectivity in both the start and end of the perturbations. The variability in the motor channels particularly the Cz is centered around the N1 peak. The FCz on the other hand has an increase in connectivity relative to other channels soon after the N1 peak as well.

In addition, the connectivity strength of the Cz, C2, and FCz channels is high

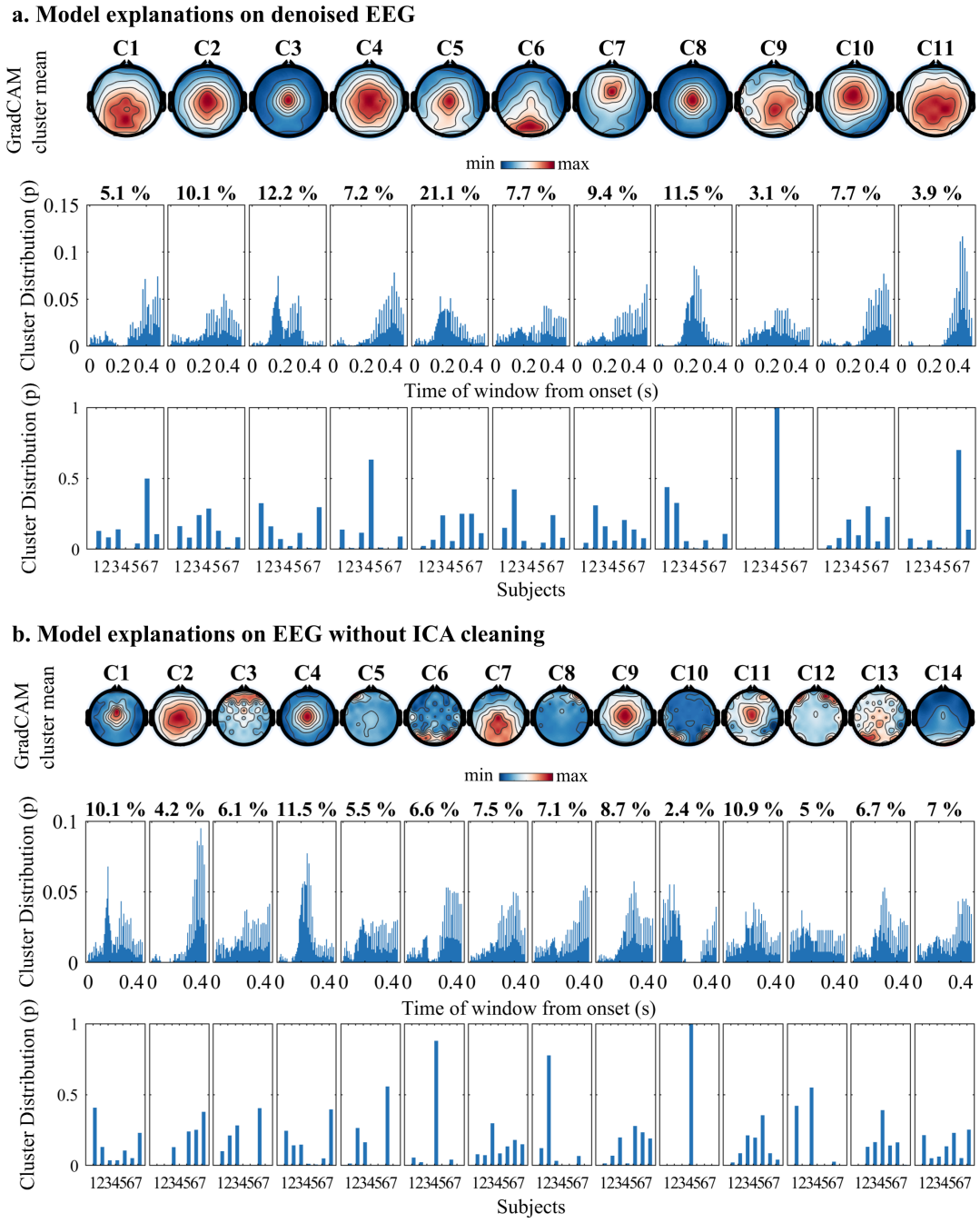

Figure 3.5: Clustering result of the model explanations from the highest performing fold.

w.r.t. the frontal and parietal channels prior to the perturbations suggesting anticipatory mechanisms. TP7 which is task-irrelevant does not appear to have significant activity throughout the duration of interest.

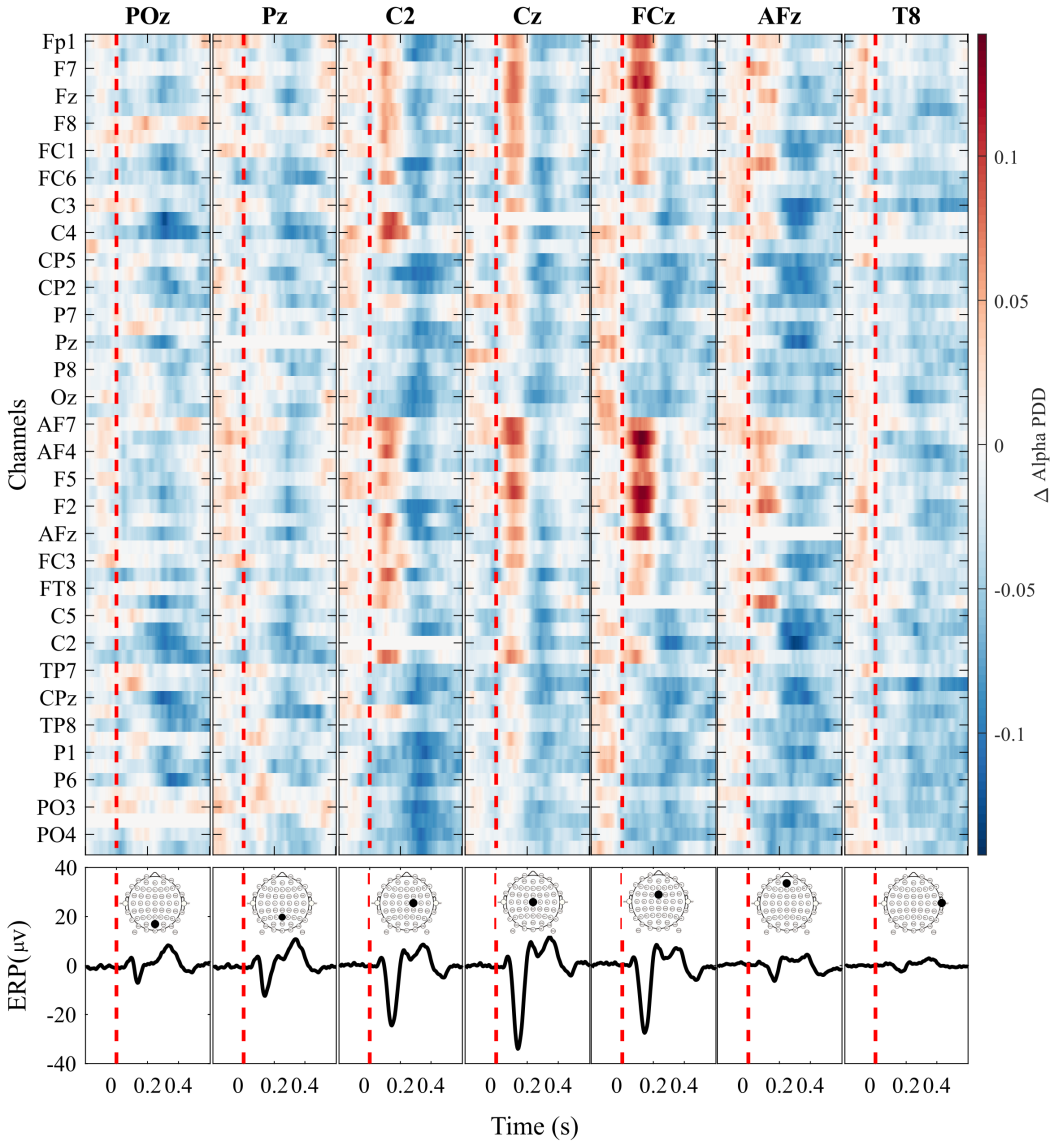

Figure 3.6: The difference in alpha band PDD w.r.t. -500 to -200 ms prior to the trials. Each column corresponds to connectivity w.r.t. one specific channel. The top row indicates how the alpha band Δ PDD of all other channels w.r.t. the channel of interest changes with time. The bottom row is a grand average PEP for the channel of interest.

3.3.4 Continuous decoding of COP from EEG

From the model explanation results and the PDD analysis, it was observed that there are dynamical changes in response to perturbations with time. Additionally, from the PEP, it is clear that distinct PEP components exist at varying latencies. With this in mind, the possibility to estimate the variation of COP associated with balance perturbation from EEG was tested. Initially, the cross-validated grid search identified the optimal hyperparameters for the GRU architecture. Fig 3.7. A) shows the distribution of R-value, R2 value, and MSE losses for all combinations of the hyperparameters used. After the hyperparameters were selected based on the performance metrics evaluated on the validation set, the optimized model was tested on the held-out test set. The performance measures are summarized in table 3.2. Evaluating the violin plot, the number of layers was found to be not critical here. The performance initially increases with the number of units but starts decreasing/saturating after 256 units. Considering this, the number of layers was chosen as one and the number of units to be 256. The model was then trained using these architectural choices.

The final optimized model yielded an across subject mean R-value of 0.7 $\pm$ 0.06, R2 score of 0.48 $\pm$ 0.1 on the test set (RTC - random timing alone), and a mean R-value of 0.64 $\pm$ 0.03, an R2 score of 0.41 $\pm$ 0.05 on the RTTC test set (random timing + type). Participant 6 had the highest decoding performance with an R-value of 0.85 $\pm$ 0.06 and an R2 value of 0.7 $\pm$ 0.4 on the test set. Participant 1 had the lowest decoding performance with an R-value of 0.45 $\pm$ 0.08 and an R2 value of 0.13$\pm$ 0.13 on the test set. Fig 3.7b. shows the continuous sample-by-sample decoder results corresponding to the best fold from the worst-performing participant (S1). Fig 3.7c. shows the continuous point-by-point decoder results corresponding to the best fold from the best performing participant (S6).

Table 3.2: GRU decoder performance metrics on the test set.

Subject	Correlation (RTC)	R2-Score (RTC)	Correlation (RTTC)	R2-Score (RTTC)
HS1	0.45 ±0.08	0.13 ±0.13	0.33 ±0.06	0.07 ±0.06
HS2	0.76 ±0.03	0.54 ±0.07	0.81 ±0.02	0.65 ±0.03
HS3	0.71 ±0.06	0.47 ±0.14	0.8 ±0.02	0.63 ±0.04
HS4	0.56 ±0.06	0.29 ±0.08	0.37 ±0.06	0.07 ±0.06
HS5	0.81 ±0.05	0.64 ±0.09	0.76 ±0.04	0.56 ±0.06
HS6	0.85 ±0.06	0.7 ±0.1	0.74 ±0.02	0.51 ±0.02
HS7	0.78 ±0.04	0.59 ±0.08	0.64 ±0.03	0.34 ±0.06
mean $\pm s.d.$	0.7 ±0.06	0.48 ±0.1	0.64 ±0.03	0.41 ±0.05

3.4 Discussion

This study investigates whether the PEP components would be preserved when a user wears an unpowered exoskeleton. It was found that all the components of the PEP were preserved and that the latency of the P1 and N1 wave preceded that of EMG and kinematic response peaks. This suggests the P1 and N1 components are a viable signal for fall prediction and prevention in exoskeletons. Fall detection in exoskeleton systems is limited and latencies are often too long to be pragmatic in real-world applications. A system detailed in Monaco et al. [114] identified balance perturbation while walking at 350 ms based on hip joint angles. It was also observed that the kinematic response from balance perturbation (while standing) peaked at approximately 350 ms. Comparatively, muscular activity peaked earlier than the COP. Also, PEP components appear as early as 75-137 ms in response to the perturbations. This provides us with a much longer window to perform actions to prevent/reduce fall-related injuries than relying exclusively on temporal kinematic features of the perturbation response.

In a recent review, Varghese et al. suggest that P1 is the earliest non-specific cortical response to a perturbation [88]. They argue that the P1 is not related to the context of the balance perturbation task, and does not contain information related

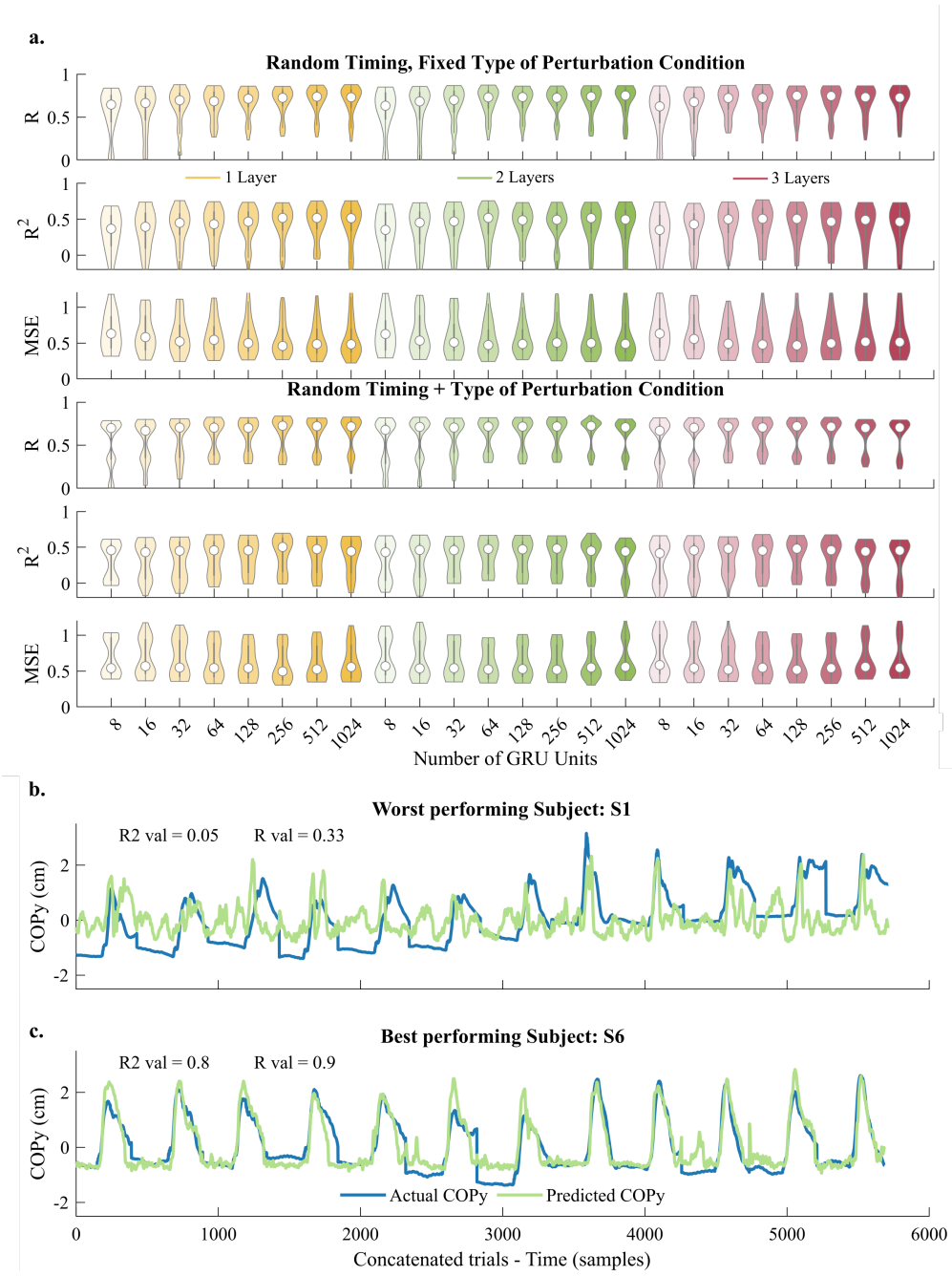

Figure 3.7: a) Performance measures evaluated on the validation set for varying hyperparameters for the GRU architectures. Each row corresponds to different evaluation metrics; b) decoded COP from the best performing subject (test set, RTC condition); c)decoded COP from the lowest-performing subject (test set, RTC condition).

to the predictability of the perturbation or whether the perturbations are internally or externally induced. It is the earliest exogenous cortical response driven by the

somatosensory input typically in the range of 0.2-12.7 μV [88]. Compared to P1, N1 is a significantly larger component distributed across the central, frontal, and parietal channels at a latency of 100-150 ms [86],[87]. Prior studies have reported the N1 peak to be as high as 60 μV, localizing in the Cz or FCz channels [86]. Unlike P1, N1 potential has been shown to not just be influenced by afferent signals. Instead, it is also influenced by the predictability and difficulty of the balance task, [129],[130] as well as the presence of competing cognitive tasks [131]. This suggests a higher-order cognitive processing role [87]. Typically, EEG data are trial-averaged to improve signal-to-noise ratio from event-related potentials. After confirming that PEP components were preserved while wearing the exoskeleton, it was determined that perturbations can also be detected from single-trial EEG. This is a crucially important step towards the real-time detection of perturbations. In real-world applications, decoding must occur in real-time. Studies decoding PEP components from single trials are rare and only find one study examining the feasibility was identified [116]. However, that particular study was conducted in a seated condition with a whole-body perturbation and did not examine standing or the use of an exoskeleton. No previous studies that target decoding PEP components from single trials in neither standing nor with an exoskeletal suit were found.

Initially, a CNN model was used to check if the presence of balance perturbation could be detected from single trials. The architecture of the CNN-based decoder was selected considering the usability of the gradCAM approach. GradCAM was chosen specifically because many of the other saliency methods were mentioned to be unreliable and GradCAM was known to be one of the most robust model explanation methods [78]. The performance was compared w.r.t the DeepConvNet [72] model. However, this study does not claim the superiority of the used model architecture or the decoder. The optimization of hyperparameters for both models was not performed, as that is outside the scope of this study. Here, the evaluation is done to confirm the existence of predictive power to detect balance perturbation on a single

trial basis and further ensure the model architecture used for prioritizing explainability is comparable to existing architectures. Subject S4 had the lowest decoding performance. During the experiment, this participant reported having congenital nystagmus. There exist a possibility that the PEP might have been corrupted by sudden eye movements and gotten removed during the pre-processing or there may be a difference in the PEP response either of which could cause a reduction in decoding performance.

It was also demonstrated that the CNN model used to detect perturbations was primarily driven by PEP components and not by artifacts. Unlike prior studies that reported few hand-selected examples to demonstrate model explanation, a clustering approach was employed in this study to visualize the model explanation of all the windows from the test and validation set. Model explanations in deep learning studies on EEG are rare. There are only a few ($\sim$ 1.5%) studies that explore the interpretability or explainability of the model used [9]. It is very important to assess whether the outputs of deep learning models are driven by artifacts or PEP signals. This is even more critical considering that the majority of the published studies using deep learning methods currently do not handle the artifacts. A recent review by Roy et al reported that only 23% of studies performed artifact handling [9]. A similar review by Craik et al. [8] reports 63 % of studies did not preprocess the EEG for classification tasks. As seen from this study, even though the prediction score is higher when using the model trained on EEG without ICA cleaning, many of the decisions were driven by artifacts.

Examining the outputs gave further confidence that the artifact handling pipeline was successful. When the model was trained on data that was not pre-processed, it was biased by artifacts as shown in figure 3.5b. The CNN started learning from bundle artifacts (C3, C6, C8, C13) and also emphasized peripheral channels more

prominently (C5, C8, C10, C12). However, these were not present when the pre-processed data was used to train the model. The study thus highlights the need for providing model explanations in deep learning studies involving EEG, as context is important to assess the main factors behind different decisions. This study also shows how using the data-driven approach coupled with model explanations can help reduce the number of channels required for the decoder. Here, the number of channels was reduced from 60 to 8 without compromising the decoding accuracy.

In addition, from the model explanations shown in 3.5a., it was observed that depending on the position of the window being considered relative to the perturbation onset (middle row), different channel combinations become most relevant. Channels in the parietal, and occipital regions were the most relevant in the earliest and the latter part of the perturbation onset (C1 and C6). Between 100-300 ms, the model shifted relevance to motor channels (C3, C5, C8). From 200-300 ms, the model was prioritizing the parietal and fronto-motor channels (C1, C2, C4, C7, C10, C11). This suggests the dynamic recruitment of different brain regions in response to the balance perturbation. The model explanations are in agreement with prior works that demonstrated the significance for these regions in balance perturbation tasks [132],[88],[133]. Further exploring these dynamics by computing a measure of dynamic functional connectivity (PDD) similar effects were observed. Specifically, the nodal connectivity was higher in the occipital-parietal region in the early stage of the perturbations, shifting to the motor, then to frontal, and back to the parietal channels.

Given these dynamics, it was expected that the EEG would have the information to be able to continuously decode the instantaneous COP variation. This was validated using a GRU model to decode continuous COP responses from single-trial EEG. It was demonstrated that the GRU model was able to decode, on a sample-by-sample basis, the COP variability from EEG alone for all participants. Evaluating the hyper-parameters, it was observed that the number of layers did not contribute significantly

to the model performance, which is in agreement with prior work [127]. However, the number of hidden layer units does impact the model significantly. This effect appears most noticeably in the variance across different folds. Comparing the three metrics, a U-shaped relationship was observed between the number of units and the decoding measures, with the performance peaking at 128 or 256 units. The variance was higher with a smaller number of units, suggesting lower predictive power in small models yields poor performance on out-of-distribution data. The variance again increased for large values of hidden units, mostly indicating the tendency towards overfitting to the training data. The selection of an appropriate number of hidden units per layer seemed to be the most critical model hyperparameter. Additional tests were conducted on similar types of trials (i.e. the backward perturbations described above) which were randomly introduced in between variable types of perturbations that included toes up, toes down and forward translations. There was a slight reduction in performance in this condition potentially resulting from additional cognitive processes required to anticipate both the timing and the type of perturbations. The decoding score across all participants exhibited good performance (R-value greater than 0.5), except for participant 1. Participant 1 consistently opted for a specific, non-stereotypical strategy to counteract the perturbation. However, it was noticed that the strategy used by this participant was not working effectively as the participant had the greatest difficulty restoring postural equilibrium. It is possible that the strategy chosen by this participant conflicted with the variable nature of the perturbation, and led to poor decoding.

In summary, relevant components in PEPs were detected as early as $\sim$ 75-137 ms after the onset of a mechanical external perturbation. These components preceded both the peak in EMG activity ($\sim$ 180 ms) and the COP data ($\sim$ 350 ms). It was observed that the perturbations could be decoded from single-trial EEG using a CNN model. Also, it has been demonstrated that the model was driven primarily by relevant components in the PEP to infer the predictions and not by artifacts. The

model explanations further aligned with the dynamic functional connectivity measure estimated using PDD. Moreover, the feasibility of decoding continuous COP values from the EEG using a GRU model was established. Overall, the findings suggest that the EEG signals contain short-latency neural information related to an incoming fall, which may be useful for developing brain-machine interface (BMI) systems for fall prevention in neurally-controlled robotic exoskeletons.

Chapter 4

Motor Imagery: through the Lens of a Convolutional Neural Network

4.1 Introduction

The ability to decode motor intent from brain activity to control external devices is the core principle behind the application of many of the Brain-Computer Interface (BCI) systems being developed. There exist different paradigms of BCI depending on the source of control signals used. Many of the signals are generated without conscious intent in response to external stimuli called evoked signals [134]. Evoked responses could be elicited by visual stimuli such as Visual Evoked Potentials (VEP, SSVEP), or sensory-based evoked responses like Somatosensory Evoked Potentials (SSEP). It could also be evoked signals in different odd ball paradigms such as P300 which occur when a participant is exposed to infrequent/odd stimuli [135]. Motor imagery (MI) on the other hand is the act of mental rehearsal of a motor action without any overt movement. MI is said to result from a 'conscious attempt of accessing the contents of intending to move which are typically done when one engaged in movement preparation unconsciously' [136]. BCI based on MI is one of the most common, yet probably one of the most difficult BCI tasks available. Studies have reported that over 15-40% of the participants are unable to control the BCI based on MI [137],[138]. This challenge of "BCI Illiteracy" still remains one of the biggest research challenges in EEG-based sensorimotor BCI [139]. There are many reasons for the inefficiency to

control the BCI using imagery. It could be state-dependent, as decoding could depend on the attention state, fatigue, frustration, or other psychological factors. [140]. Unfamiliarity with the technology could be another factor as participants would need to become familiar with and learn the internal model of the external device being controlled by the BCI in the first place [141],[142]. For another group of participants, the cause could also be physiological. There exist the possibility that the neuronal population contributing to the motor imagery could be localized in the folds of the brain and the sensors on the scalp would be unable to pick them up [143]. However, one important reason which is not emphasized enough is the sensor configuration, neural features, and the decoder itself that is used for the decoding purpose. Considering that targeted signals in most decoders are sensorimotor rhythms, typically the EEG sensors and the decoders localize on the sensorimotor region for training the decoders. A recent review paper looking into deep learning studies on MI also identified that majority of the studies do localize on the sensorimotor region [10]. Fig 4.1, summarizes some of the most commonly used configurations in the studies. BCI competition is one of the most popular MI-based BCI datasets currently available. BCIC IV 2a [144] localizes in the sensorimotor region in a diamond configuration and occupies 37 % of MI studies that use deep learning. Similarly, BCIC IV 2b [145] and BCIC II [146] use a 3 channel configuration and together occupy an additional 43% of studies. A significant proportion of studies limit their channel to the configuration resembling the third column [10]. This assumes that when the participants are directed to perform motor imagery, everyone recruits a single strategy to achieve that. However, this might not always be the case. It could also be due to communication problems between the experimenter and the participant. Not being able to articulate the requirements of the BCI well or how they respond to the instructions could cause variability in how they engage with the BCI.

To address some of the challenges above, this study proposes a data-driven approach to study the underlying brain dynamics of motor imagery. Here, without

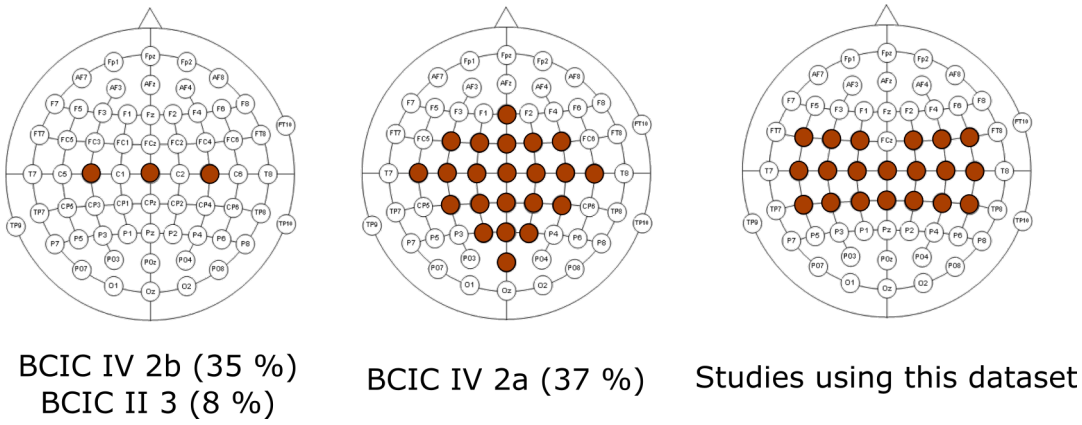

Figure 4.1: The channels used in the majority of deep learning studies looking into motor imagery. The percentage values are obtained from [10].

hand-selecting the channel configuration or the neural features, whether the deep learning model can identify the different strategies used by individuals were tested. Later, how these model explanations can improve the BCI decoders were also evaluated.

4.2 Methods

4.2.1 Dataset

For replication purposes and to include data from a large sample of subjects, the MI data collected by Lee et al. [147] was used in this study. It consisted of 62 channel EEG sampled at 1000 Hz, collected from fifty-four healthy individuals (age: 24-35 years, 25F). Thirty-eight subjects were naive BCI users whereas the others had prior experience working with BCI experiments. The data was collected using the BrainAmp system (Brain Products, Munich, Germany) referenced to the nasion, and grounded with respect to the AFz channel. The electrode impedance was kept below 10 Kohm. Each trial consisted of 3 seconds of looking at a black fixation cross at the center of the screen. Then an arrow appears for 4s pointing left or right directing

the participants to perform hand imagery by imagining grasping the respective hand. Each trial ended with a rest period of 6 ± 1.5 s. A total of 100 such trials on two separate days/sessions were recorded. In the original data, they trained a model on the training set and tested it with real-time visual feedback during a test set that included 100 additional trials. To avoid confound related to decoder-specific BCI used in that study, the test trials are not included here. The experimental design for an individual trial is summarized in Fig 4.2

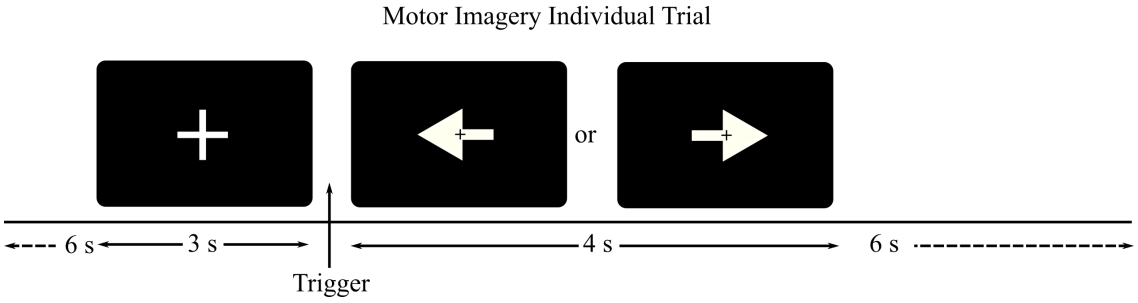

Figure 4.2: Motor Imagery experimental design for an individual trial.

4.2.2 Pre-processing

The 62 channel EEG was initially down-sampled to 250 Hz for computational efficiency. Later the signals were high pass filtered using a 4th order Butterworth zero-phase filter to reduce the drift artifacts. The cutoff frequency was set to 0.3 Hz. The data was then fed into an IIR notch filter with a Q-factor of 20 to remove 60 Hz line noise. The EEG was then decomposed into the independent components using the Infomax algorithm [148]. The artifactual IC's were then removed an automated process using the ICLabel toolbox. The thresholds for rejection were as follows: Ocular (60%), Muscule (50%), Heart (70%), Linenoise (70%), Channel noise (60%), or if the identified percentage for Brain is < 10 %. After denoising the EEG per session, the trial data is Z-scored w.r.t. to 6s window prior to each trial. Later, the continuous trials data during the MI task is segmented into 1.5 s long windows

with an overlap of 40 ms. All the pre-processing steps were implemented using the EEGLAB library[121]. The pre-processing flowchart is summarized in Fig 4.3

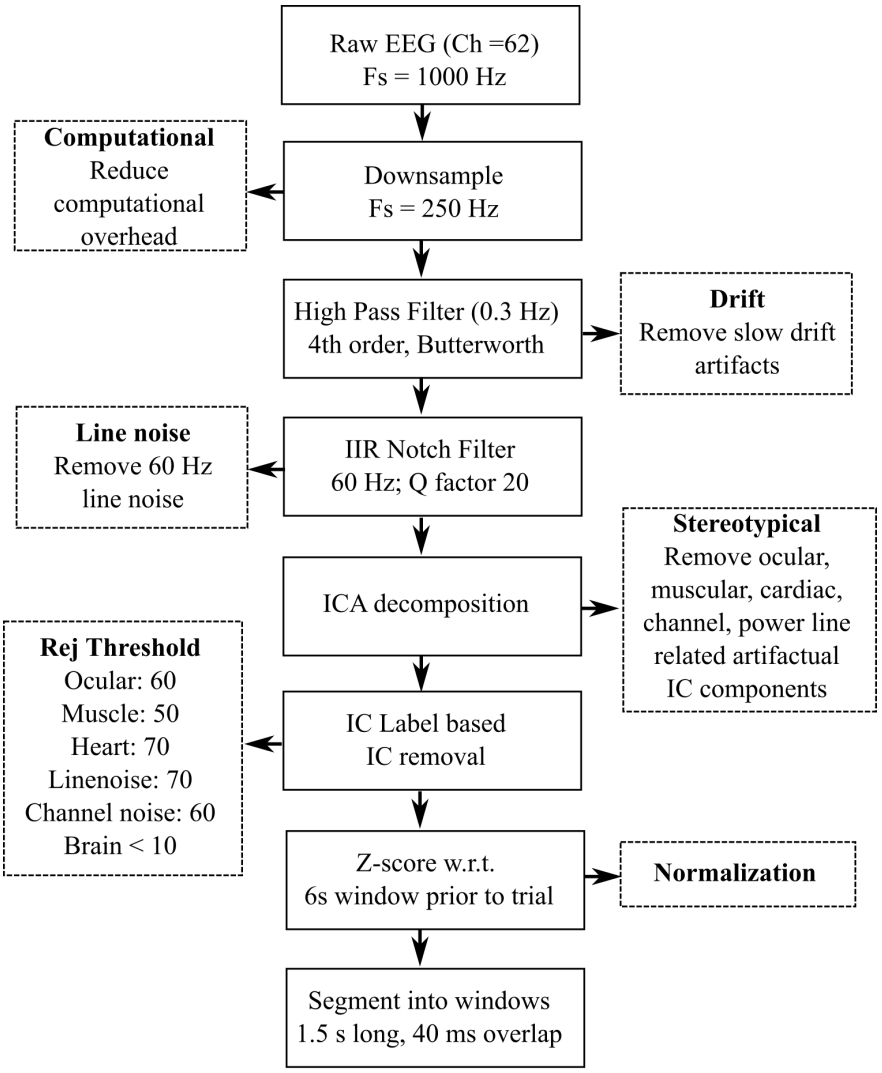

Figure 4.3: The pre-processing flowchart used to remove the artifacts and prepare the data for classification.

4.2.3 Convolutional Neural Network

The architecture for the model is summarized in Fig. 5.3. The input to the model is the 1.5 s EEG window (batch size x 375 samples x 62 channels). The model consisted of 5 temporal convolution layers of 32 units each (5 x 1 kernel size with

a stride length of 1). A temporal pooling layer of 2x1 pooling dimension with a stride length of 2 was also used after every convolutional filter layer except the last two blocks. The output from these convolutional layers was flattened and fed into a dense, fully connected layer of 32 hidden units followed by an output layer with softmax activation.

A dropout layer with alpha = 0.5 was added in between the dense layer and the output layer to reduce overfitting. Except for the output layer, the model utilized ReLU as the activation function. The proposed model was implemented in python 3.7 using Pytorch library [81].

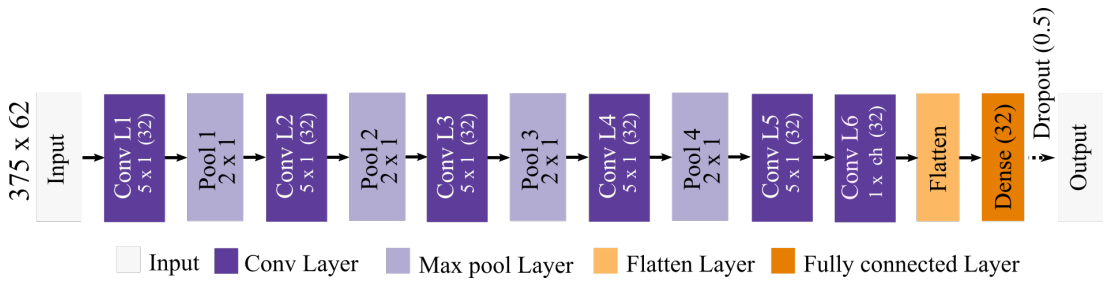

Figure 4.4: Model architecture: Each block correspond to different types of layers in the model. The dotted line is to illustrate the dropout operation during the training phase aimed at reducing over-fit. During inference, all units were retained.

4.2.4 Training the CNN

The CNN was trained to classify between left hand and right-hand motor imagery. A 5-fold cross-validation was performed to estimate the mean decoding performance. A re-initialized independent copy of the same model architecture was used for each fold on a per subject basis. An Adam optimizer [122] with a learning rate of 0.001 was used to train the model. A batch size of 128 and epoch length of 100 was set. An early stopping condition was set to avoid the model from being over-fitted. This stopped the training if the validation loss did not improve in 10 consecutive epochs.

To ensure reproducibility and consistency Numpy, Pytorch and Cuda random number generator were all seeded by the fold number.

4.2.5 Model Explanation

To identify the segment of EEG the CNN looked at to arrive at the correct predictions pertaining to each class of interest, Deeplift based model explanation was applied for all correctly predicted data points in the validations set [35]. Deeplift assigns the relevancy to the input data points by backpropagating the contribution of the output activation to the input and comparing the activation w.r.t. a reference set and the relevancy is assigned as a function of this difference w.r.t. the reference. Here, the reference/baseline input used is an input of zeros as was done by Lawhern et al. 2018 [65]. Once the model explanation was extracted from each of the data points, instead of hand-selecting individual data points, a clustering approach to pool the model explanations from all the participants was used. Initially, the individual explanations were time-averaged to get the scalp relevancy maps. Then, to make the explanation's scale comparable, the data was normalized by dividing by the maximum absolute relevancy per explanation. Later, the normalized explanations from all the subjects across all the folds from the validation set were combined to be used for clustering using k-means clustering. The process is summarized in Fig 4.5.

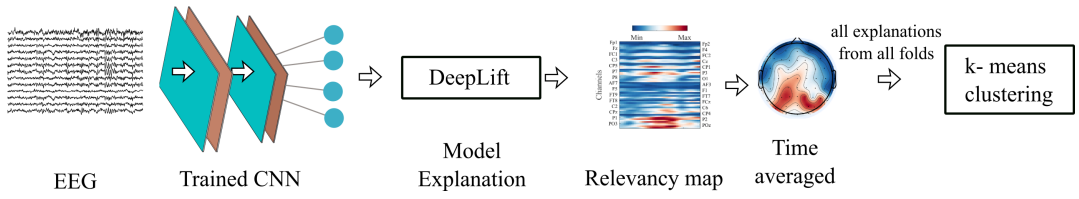

Figure 4.5: Clustering flowchart: The figure summarizes the clustering flowchart to pool the explanations across subjects.

Clustering was done separately for each of the classes. To estimate the ideal cluster number, the k-means were estimated iteratively from K = 1 to 54. Instead of

manually selecting the K values, the Kneedle algorithm[125] was used to estimate the Knee of the curve. The knee was extracted from the curve corresponding to the mean within-cluster distance for each of the k values. The knee point was then identified by the Kneedle algorithm. The process was repeated 10 times and the mean K value was selected for the final clustering.

4.2.6 Impact of channel selection

To evaluate the impact of channel selection, a separate model looking at EEG from a subset of channels in the sensorimotor region as in Fig 4.1.c was trained. The channels selected were FC5, FC1, FC2, FC4, FC6, C5, C3, C1, C1, C2, C4, C6, CP5, CP3, CP1, CP1, CP2, CP4, CP6. The change in accuracy w.r.t. using all the channel montage was then compared.

4.3 Results

4.3.1 Individual subject model training

For the individual intra-subject model trained on each subjects data alone, a mean cross-validated test accuracy of 60.7 $\pm$ 10.2 % was obtained with subject 36 having the highest accuracy of 89.3 $\pm$ 3.7 % and subject 34 had the lowest accuracy of 46.7 $\pm$ 4.8 %. The chance level was 50 % as the cross-validation split was done ensuring an equal number of trials and windows from both classes were preserved in all the folds and sets. The distribution of decoding performance for each individual subject across the folds arranged in increasing order of mean test accuracy is summarized in Fig 4.6

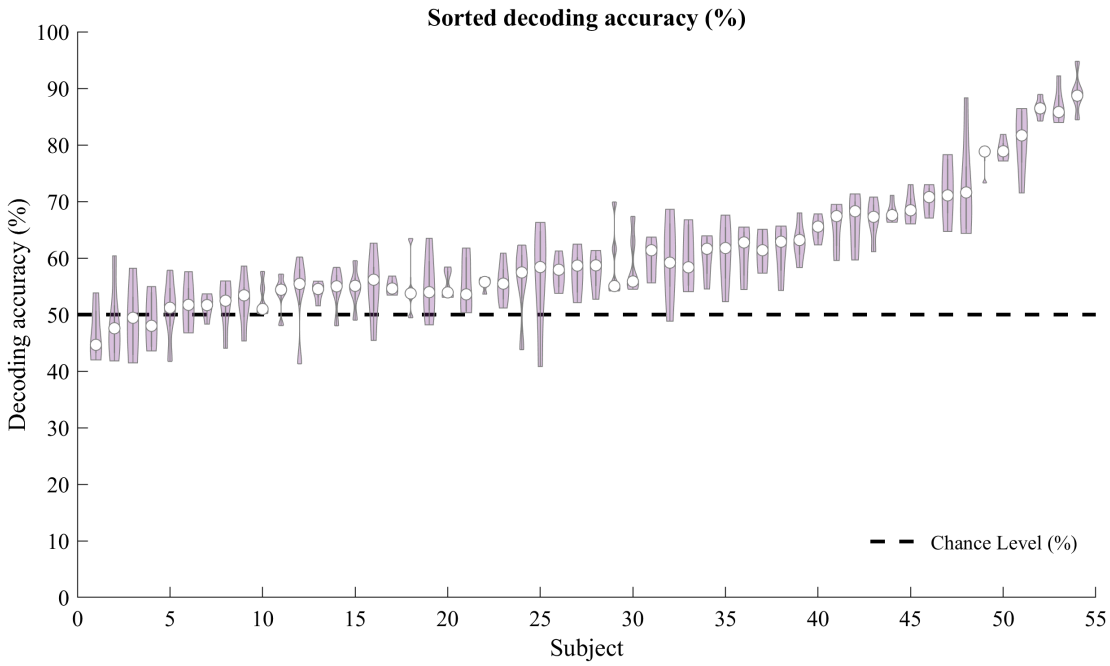

Figure 4.6: The distribution of test accuracy across the 5 folds for each individual subject, arranged in increasing order of performance. 50 % is the chance level of decoding.

4.3.2 Model explanation analysis

The optimal K value was identified to be 8 per class. The clustering results are summarized in Fig 4.7. The top two rows correspond to the right-hand motor imagery and the bottom two rows correspond to the left-hand motor imagery. The histogram is sorted in increasing order of accuracy. Each bin corresponds to the percentage of data from the particular subject in that cluster. The bin in the right would correspond to the top-performing subjects and the bin in the left would correspond to the low performing subject. Cluster 6 in the RH and cluster 5 in the LH group is localizing in the channels over the motor cortex, which is over the area of the cortex with hand representation. It is localizing the region and by further evaluating the histogram, it

was found that subjects in that cluster are also the top-performing subjects. Similarly, C1 (RH), C7(LH) is also focusing on the channels on the sensorimotor region.

However, it was observed that multiple clusters primarily localize in non-motor regions like C2, C4 in LH, and C2, C4 in RH. These are focusing on either the parietal or the occipital channels or both. Cluster C8 on the other hand seems to be focusing on the temporal channels and they correspond to some of the low-performing subjects.

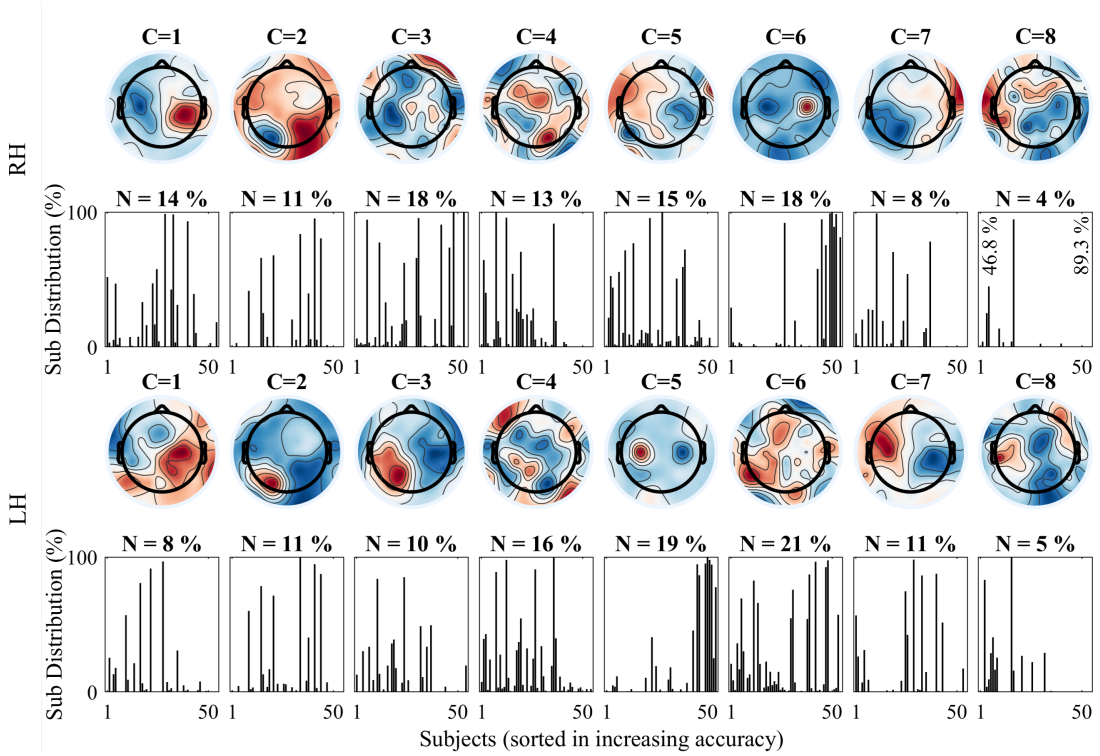

Figure 4.7: Cluster representation summarizing the different subset of network configuration the model focused to arrive at the correct prediction for each of the classes. The histograms are sorted in increasing order of decoding accuracy. They represent the percentage of each subject that belongs to that particular cluster.

4.3.3 Impact of channel selection

Figure 4.8 shows the distribution of the difference in decoding performance when using the motor only channels w.r.t. using all the channels.

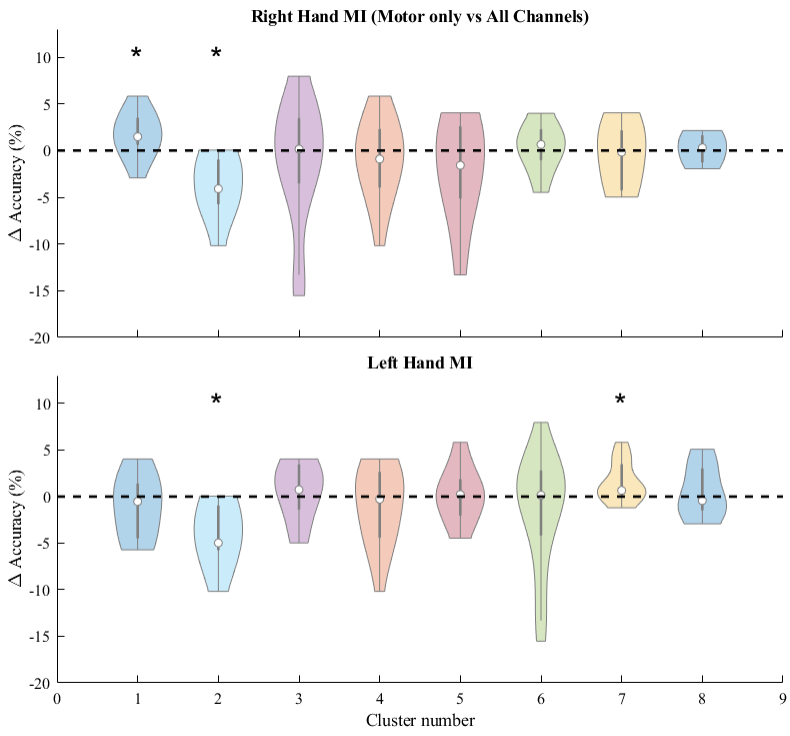

Figure 4.8: Distribution of difference in decoding performance for when using sensorimotor montage vs using all the channels for each cluster. * indicates statistically significantly different from zero for $p < 0.05$.

Each violin plot corresponds to this distribution in the respective cluster. The subjects are considered only if the cluster contains at least 12.5 % of their data which is the chance level for 8 clusters. For subjects who are in cluster 2 in either of the class, the decoding is statistically significantly and consistently lower ($p < 0.05$) if the motor channel montage alone is used (RH: $p = 0.008$, LH: $p = 007$). On the other hand, subjects in cluster 1 in RH and cluster 7 in LH perform better when using the motor region montage. The difference in accuracy for the subjects when using motor channels alone in these clusters were statistically significantly higher compared to using all the channels for $p < 0.05$; C1 (RH): $p = 0.02$, C7 (LH): $p = 0.04$. Cluster 4 and 5 for RH and which had relevant channels outside of the motor-only channels had a lower median accuracy compared to using all channels. However, it was not statistically significantly lower.

4.3.4 Cluster specific training

To further evaluate whether the model explanations could be used to improve the decoding performance of individual subjects, the clusters that localized in the motor channels (C6- RH and C5 LH) were selected. Later, subjects whose 50% or more data belonged to either of these clusters (N = 10) were identified. A separate reinitialized model with the same configuration was pre-trained using the data from all these selected subjects. Later the pre-trained model was fine-tuned for each subject with > 12.5 % of their data represented in the cluster (N = 15). Evaluating the decoding performance, it was observed that doing the fine-tuning significantly improved the decoding compared to training on individual subjects ($P < 0.05$) as depicted in Fig 4.9c.

It was important to know if the decoding will be good if the base model was trained on 10 subjects selected randomly or if it is specific to pre-selecting the subjects. Training on randomly selected subjects had a median decoding performance drop and was significantly lower than cluster-specific training ($P < 0.05$) fig 4.9c. Later, it was evaluated whether training on all the subject's data would improve the decoding similarly. It was found that doing so did end up significantly improving the decoding w.r.t. training on individual models but the median accuracy was still lower compared to training on cluster-specific approach. However, the difference between using all the subject's data and cluster-specific training was non-significant at $p < 0.05$. To evaluate the possibility that the non-significant reduction in decoding could be caused by the inclusion of very low performing subjects upon training on all subjects data, the process was repeated by only including subjects with accuracy $> 60\%$. Interestingly, it was found that doing so led to a further drop in decoding accuracy. This suggests the relevance and need for identifying participants who share similar strategies and using that to selectively train the models. To further evaluate the performance difference with more detail, the difference on a subject-by-subject basis

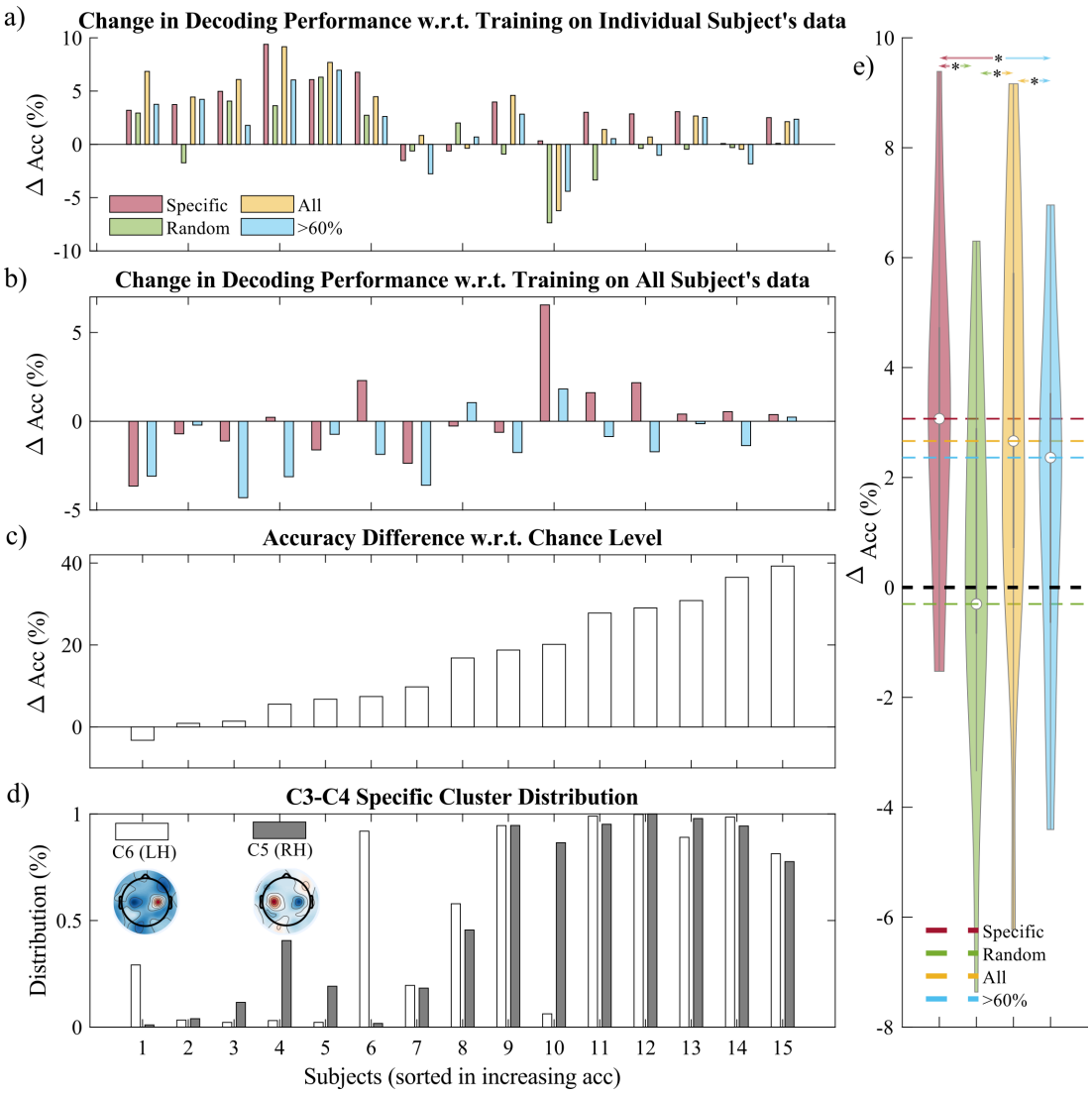

Figure 4.9: Performance difference w.r.t different pre-training strategies. a: difference in decoding performance w.r.t. training on the subjects data alone; b: difference in decoding performance w.r.t. training on all the subjects data; c: baseline difference in decoding performance for each individual arranged in increasing order of accuracy; d: distribution of data from each subject in either of the selected clusters; e: distribution of change in decoding performance w.r.t. training on individual subject's data alone.

was evaluated. Fig 4.9a-d gives a detailed illustration of a performance differences in each of the subjects in the cluster. Here, using either all the subject's data or cluster-specific training improves training in most subjects. Interestingly, compared to either using all the subject's data, or the subjects with >60% decoding accuracy, using the

cluster-specific approach is consistently better for all the top-performing subjects. For the low-performing subjects, both the cluster-specific approach and using all the subject's data still improves the decoding compared to training on individual subject data. However, using the data from all the subjects seems to work better for these participants than the cluster-specific approach.

4.4 Discussion

BCI provides a means of interaction with an external device or to communicate using brain activity. However, prior studies have shown these do not work for all subjects often leading to a subset of individuals being labeled as "BCI illiterate". Previous studies have shown that various solutions such as improved signal processing, additional feedback sessions, more number of sessions, providing instructions differently etc have resulted in improving decoding [143] in people who are labeled under the "BCI illiteracy" category. However, currently, the different strategies used are applied to all the individuals purely based on decoding performance value. This study offers a possible alternative- by using the model explanation approach, a data-driven approach can be used to identify different strategies or a subset of individuals. Using this information, BCI illiteracy could be reduced. Model explanations could throw some insights into why some individuals perform better compared to others, why some respond to specific feedback training whereas others do not etc. This study identified that for some of the subjects the most relevant channels are not in the sensorimotor regions perhaps indicating the use of alternative non-MI strategies. This is important as most of the existing BCI on MI extract specific features from a subset of channels in the sensorimotor regions. When individuals are labeled as BCI illiterate based on an arbitrary set decoding threshold [138], at least some of them may fall under this category, as they might be eliciting a different strategy/channel configuration. This was observed in our analyses as well with subjects having relevancy

localized in the occipital-parietal regions, their decoding is consistently lower when using channels in the motor cortex compared to using all the channels. Some of them might be using visual imagery instead of motor imagery. A study by Stinear et al. showed that corticomotor excitability is modulated only in kinesthetic imagery and not visual [149]. Similar a study by Neuper et al demonstrated that visual motor imagery did not elicit any clear spatial pattern in the sensorimotor hand area [150]. When only performance accuracy or a particular signal or interest (e.g. ERD/S in the motor channels [151]) is considered, the absence of signal or low decoding could partially be accounted for by the fact that they could be using a different strategy. Instead of labeling them to be BCI-illiterate, trying to personalize and diagnose the reason for poor performance would be critical to the advancement of BCI adoption and development. Using model explanations in a data-driven manner as done here provides more context than just a simple number to cluster individuals. This would allow a better evaluation of the effectiveness of different types of interventions and decoders.

Evaluating the cluster explanations, it was identified that some of the model explanations were localized in the inferior parietal regions. In a lesion study, Schwoebel et al. showed that people with parietal lobe lesion was unable to predict the sensory outcome or the time required to complete the hand movement. They were unable to prevent overt movement during the motor imagery [152]. Another study involving TMS stimulation showed that the right inferior parietal lobe (rIPL) conditioning 6 ms prior to M1 stimulation facilitated the motor evoked potentials (MEP), whereas the facilitation was abolished during mental rotation. Similarly, the corticimotor excitability was suppressed during MI [153]. This suggests the parietal lobe play the role of movement inhibition and in terms of motor imagery that reflects to preventing the hand from moving during imagery. For some of the participants with distribution prevalent in the cluster localizing parietal and non motor channels, they might be engaging in a strategy to avoid hand movement, than focusing on the kinesthetic

imagery. Follow-up questionnaires and experimental design based on the model explanations might shed more light on the strategies used by these individuals.

This study also showed that pooling individuals with similar scalp relevancy and pre-training the subject independent model trained on this subset of individuals improved the decoding performance of the subjects. The decoding performance was significantly better than training on a random subset of the same number of subjects. It was found that the training on data from all the subjects had lower median accuracy and also higher variance but was not significantly different. Assuming this to be caused by the inclusion of very low performing subject's data bringing the decoding lower, the model was retrained using data from subjects with 60% and above decoding accuracy alone. Interestingly, this brought the decoding even lower. This suggests the improvement is not associated with removing the low-performing subjects but the potential benefit of identifying subjects with similar strategies prior to pooling. Later, to better understand the cause of non-significant reduction in decoding when using all the subject's data compared to subject-specific approach, the change in performance at an individual level was explored. Here, it was observed that the top-performing subjects consistently performed better when using the cluster-specific fine-tuning compared to using data from all the subjects. However, the low-performing subjects who also have a low percent of points in those clusters benefit from pooling data from all the subjects. Even though the performance improved in them while using cluster-specific approach, the accuracy was better when using all subject's data. This could come from the fact that windows corresponding to the mean cluster distribution was low in these subjects and they might not have a strong consistent pattern. For them, using all the subjects could help them learn more variable representations eventually providing additional benefits. On the other hand, doing so in top-performing subjects, who have a clear strategy would have the opposite effect. Since they have a clear and strong signal, adding variability and subjects using different strategies would be detrimental and confuse the decoder. For them, they would prefer to have subjects

who share a very similar strategies. Therefore, using model explanation, personalized model improvement especially for high-performing subjects could be done.

In conclusion, this study shows how model explanation could identify the underlying scalp representation of motor imagery in a data-driven manner. Without providing any prior domain knowledge, the model was able to localize channels in the sensorimotor region to be the most critical in decoding. They also identified a subset of individuals who did not engage these regions whereas they clearly localized channels in the occipital-parietal regions. The model explanation could be used to identify potentially different strategies used by individuals and how limiting channel montage to the sensorimotor region might not be ideal for all participants. Moreover, the clustering approach using model explanation improved the decoding in high-performing subjects. Overall, the explainability approach can lead to personalized measures of handling BCI illiteracy and potentially provide insights into why specific individuals respond to certain interventions compared to others.

Chapter 5

Susceptibility of Deep Learning based BCI Decoders to Eye Blink Artifacts

5.1 Introduction

Brain-Computer Interfaces (BCI) are systems that allow an individual to control different end effectors using his/her brain activity. Electroencephalography (EEG) is a commonly used modality for non-invasive control of BCI. These systems typically involve a multi-stage process starting with data acquisition, followed by pre-processing to clean artifacts. The task-specific features are then hand crafted from these de-noised signals which are then fed into a classical machine learning model. With the recent advancement of computational tools available, the decoders used to develop such systems have been replaced with more sophisticated yet data-driven models like convolutional neural networks and other deep learning models. These models have been shown to improve the state-of-the-art decoding performance compared to regular linear models. However, a significant amount of studies employing deep learning, currently do not address the artifacts present in the data. The total percentage of studies as high as 67 % [8] to 77 % [9] either do not handle artifacts or report whether they de-noised in their studies. This is even worse in studies focused on motor imagery being as high as 85 % [10]. This is further exacerbated by the fact that less than 1.5 % of studies [9],[8] employ some form of interpretability in their analysis making it difficult to assess whether the model has learned to avoid artifacts or not. There exist some studies that use deep learning either to detect artifacts or

solely artifact removal [154]. However, there are no systematic studies that evaluate how a deep learning model trained on an end-to-end basis handles artifact. One commonly observed artifact which would be present independent of the task is eye blink artifacts. This is particularly important and relevant since blink artifacts in EEG are typically much higher amplitude compared to the background brain activity. Considering CNN's are data-driven model that can be thought of as a model that looks for distinct patterns in the data, how they handle these high amplitude events is surprisingly not discussed in the literature.

This study uses synthetic (simulated) eye blink data to systematically vary the rate of eye blinks added to single-trial EEG data. To simulate the eye blinks, many of the prior studies have made use of simple exponential functions to approximate the blinks. However, these do not capture the complete morphology of the blinks. Considering this, the researchers used real blink data as a source-level signal and forward projected them using the leadfield matrix using the SEREEGA toolbox [82]. This allows us to precisely control the proportion of blinks in the trials and how they bias each class. Even though the current analysis is limited to eye blinks, the proposed framework could be used to study the impact of different kinds of artifacts.

5.2 Methods

5.2.1 Simulation

The flowchart for generating the simulated eye blinks is shown in Fig 5.2. The blink waveform was extracted from the EOG data collected from the BCI competition dataset [145]. The EOG was collected from the nasion location sampled at 250 Hz from a total of 9 subjects. Each subject participated in a total of two sessions recorded on two separate days. Each session consisted of a total of 120 trials of MI. This dataset

was used only for extracting some real blink waveforms to be added as source signal in the SEREEGA simulation framework [82]. The data was initially band pass filtered between 1-30 Hz. An IIR notch filter at 50 and 100 Hz was employed to remove line noise and its harmonics. In some datasets, there were a few bad segments consisting of high amplitude deflections which were initially removed manually. To reject outliers in peak detection, blinks were screened by passing the data through the blinker algorithm [155]. The algorithm compares different morphological attributes of blink to reject outliers or non-blink peaks. The filter parameters in the algorithm were set to 1-20 Hz filtering, blinkAmpRange set to [3,20], blinkfits correlations greater than at least 0.99. Later, 3.8 s long EOG (same as MI trial window used), centered around the blink was extracted. The latency of the blink inside the window was sampled from a normal distribution centered at 1.2± 0.2 s. In addition, adjacent blinks separated by at least 1.5 seconds were only retained to ensure maximally one blink is present in each window and that blinks do not overpower the window. The blinks were normalized by the positive peaks and windows of blinks wherein the maximum negative deflections are greater than 0.5 were also removed to reject any outliers. The extracted waveform was then used as an activation signal for dipoles to simulate eyes. The signal from these dipoles would then be forward projected to the scalp to replicate blinks.

Since none of the existing leadfield matrices include dipole locations outside the brain or replicate the eye locations, 3 dipoles were selected such that when the activity of the dipole was projected, the scalp projection simulates that of eye blink independent components. Multiple MRI images including the eye were studied in order to place the dipoles as close to where eyes are supposed to be present. The dipole locations were the closest one to the MNI coordinates (-80,90,-60), (0,90,-60) and (80,90,-60) with orientation set to (0,1,0). As can be seen from 5.2, the forward projected scalp distribution resembles eye blink independent components typically found using ICA with weights being higher on the frontal channels and reducing from front to back.

5.2.2 Dataset

For replication purposes and to include data from a large sample of subjects, the MI data collected by Lee et al. [147] was used. It consisted of 62 channel EEG sampled at 1000 Hz, collected from 54 healthy individuals (age: 24-35 years, 25F). Thirty-eight subjects were naive BCI users, whereas the others had prior experience working with BCI experiments. The data was collected using the BrainAmp system (Brain Products, Munich, Germany) referenced to the nasion and grounded with respect to the AFz channel. The electrode impedance was kept below 10 Kohm. Each trial consisted of 3 seconds of looking at a black fixation cross at the center of the screen. Then an arrow appears for 4s pointing left or right directing the participants to perform hand imagery by imagining grasping the respective hand. Each trial ended with a rest period of 6 $\pm$ 1.5 s. A total of 100 such trials on two separate days/sessions were recorded. In the original data, they trained a model on the training set and tested it with real-time visual feedback during a test set that included 100 additional trials. To avoid confound related to decoder-specific BCI used in that study, the test trials were not considered here.

The experimental design for an individual trial is summarized in Fig 5.1

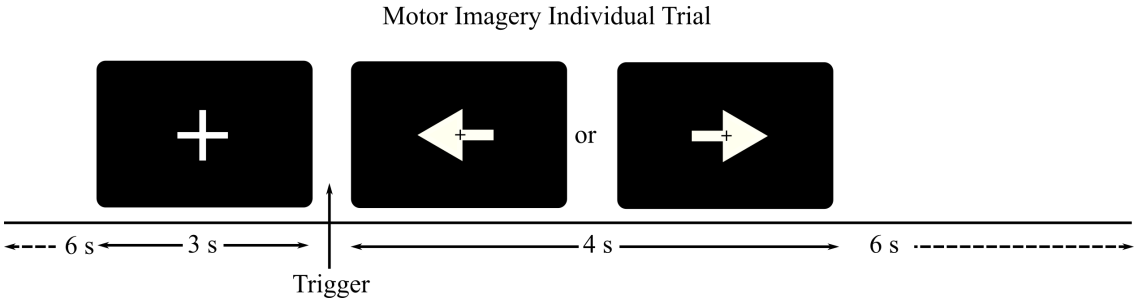

Figure 5.1: Motor Imagery experimental design for an individual trial.

5.2.3 Injection of blinks in varying proportions

To evaluate how the frequency of blink artifacts in the trials impact the decoding performance, simulated blinks were randomly assigned to varying percentages of trials. To also evaluate the impact of class imbalance, the process was repeated with and without the blinks biasing a particular class. Three different proportions of trials were tested. In the balanced class condition, the number of trials with blinks was made equal across the two classes (10%, 50%, and 90% of trials). This helps understand how the frequency of these high amplitude artifacts (even when they do not add any class-specific information to the decoding) would impact the CNN performance. The impact of blink artifacts was also tested in the unbalanced condition wherein blinks are more in one class than the other. For class 1, blinks were added to 20%, 50%, and 90% of the trials. Additionally, blinks were also added to random 10% trials of class 2. A paired t-test was then done to evaluate the performance difference for each of the participants with and without the blinks.

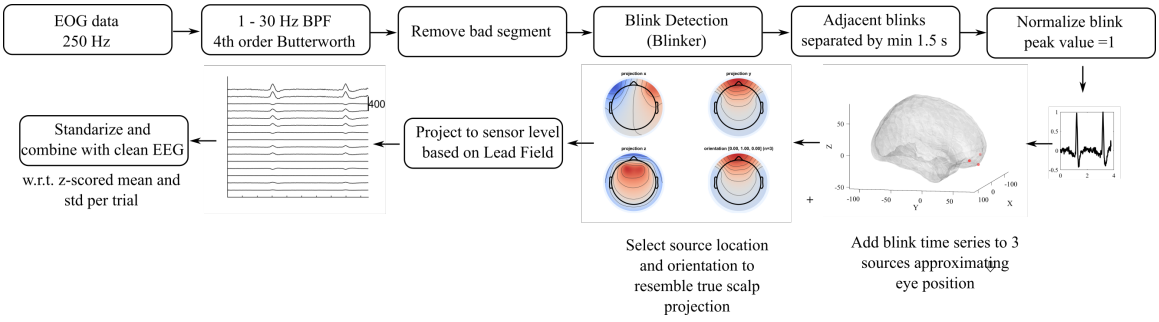

Figure 5.2: The flowchart details the process by which simulated blinks were generated with the help of the blink template and SEREEGA framework.

5.2.4 Convolutional Neural Network Architecture

The architecture for the model is summarized in Fig. 5.3. The input to the model is the 1.5 s EEG window (batch size $\times$ 375 samples $\times$ 54 channels). Eight channels

were removed as they are not contained in the forward model. The model consisted of 5 temporal convolution layers of 32 units each (5×1 kernel size with a stride length of 1). A temporal pooling layer of 2×1 pooling dimension with a stride length of 2 was also used after every convolutional filter layer except the last two blocks. The output from these convolutional layers was flattened and fed into a dense, fully connected layer of 32 hidden units followed by an output layer with softmax activation.

A dropout layer with alpha = 0.5 was added in between the dense layer and the output layer to reduce over-fitting. Except for the output layer, the model utilized ReLU as the activation function. An Adam optimizer [122] with a learning rate of 0.001 was used to train the model. A batch size of 64 and epoch length of 100 was set. An early stopping condition was set to avoid the model from being over-fitted. This stopped the training if the validation loss did not improve in 10 consecutive epochs. A re-initialized independent copy of the same model architecture was used for each fold, condition, and subject. The proposed model was implemented in python 3.7 using Pytorch library[81]. A separate independent model was trained to predict left hand vs right-hand motor imagery from single-trial EEG window for each of the participants and each of the condition (different proportion of blinks per class).

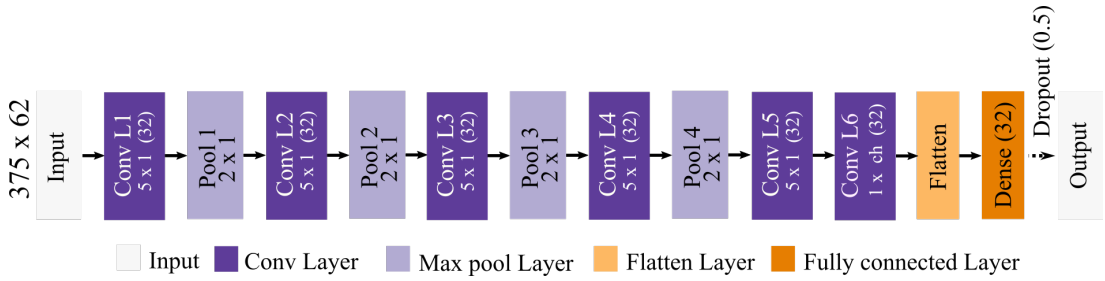

Figure 5.3: Model architecture: Each block correspond to different types of layers in the model. The dotted line is to illustrate the dropout operation during the training phase aimed at reducing overfit. During inference, all units were retained.

5.3 Results

5.3.1 Impact of frequency of blinks to decoding

Fig 5.4 summarizes the impact of varying frequency of eye blinks on the decoding performance. Each violin plot corresponds to each of the individual conditions (different proportion of trials with eye blinks in both balanced and unbalanced conditions). As expected, in the case of blinks biasing a particular class, it was observed that the decoding is significantly higher compared to training on de-noised EEG without blinks ($p < 0.05$). As the number of trials with the blinks increase, the decoding performance increases proportionally. On the other hand, in the case wherein the blinks are balanced across classes there is an insignificant difference in decoding performance when the number of blinks is low. However, when the number of blinks becomes more frequent, the decoding performance becomes significantly lower compared to data without blinks ($p < 0.05$).

5.3.2 Change in decoding as a function of original decoding accuracy

The difference in accuracy in the presence of blinks was compared with respect to the subject's original decoding accuracy in the absence of artifacts. Figure 5.5 summarizes the correlation in both the balanced and unbalanced cases. Interestingly, the subjects with higher decoding performance tend to be minimally influenced by the eye blinks even in the case of them biasing a particular class. As a matter of fact, for some of the subjects, the slight imbalance (10%) causes a reduction in decoding performance in some top performing subjects. The subjects with the poorest decoding always benefit from the blinks when they bias the class. However, when the blinks become too frequent (90% of trials), even the subjects with the higher decoding

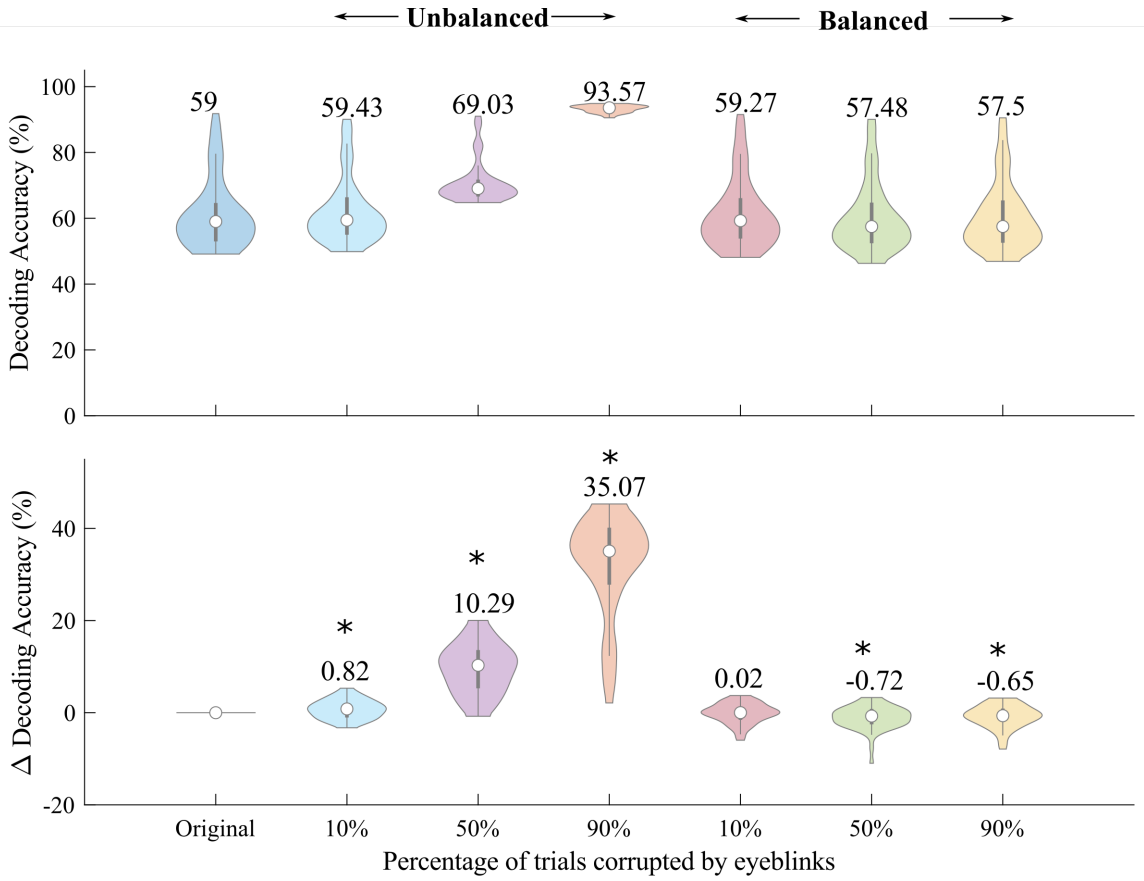

Figure 5.4: Violin plot corresponding to the mean change in decoding performance as function of both frequency of trials corrupted by blinks and the class imbalance.

performance started to make use of blinks to improve their decoding performance.

On the other hand, in the class imbalanced case, frequent blinks negatively impact particularly the top-performing subjects causing their accuracy to drop. In the case of less frequent occurrence (10 % trials), the top-performing subjects are minimally impacted by the blinks.

5.3.3 Visualizing the influence of eye blinks for the decoding

To further evaluate how the blinks are affecting the decoding, DeepLift based model decision explanation was used. Fig 5.6 shows how each of the 6 models would

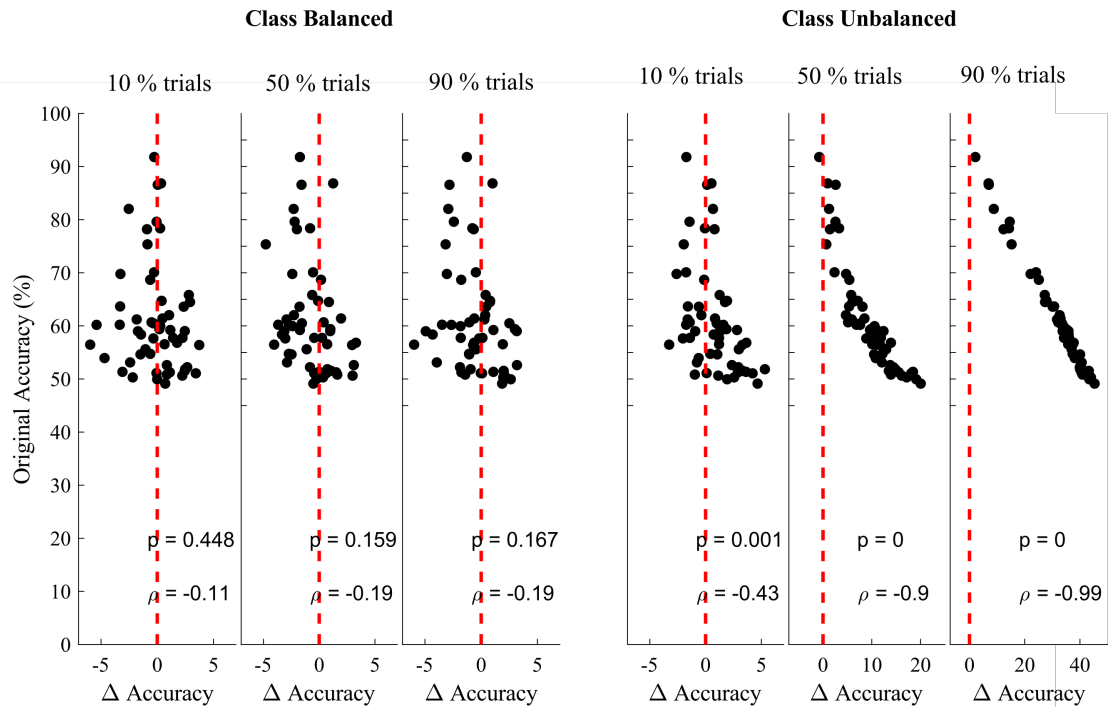

Figure 5.5: Correlation plot showing the change in decoding performance as a function of the subject's original decoding performance in the absence of artifacts.

look at the same input data for the best and the worst-performing subject. For the best performing subjects, the model is focusing on features centered around differences in alpha band power between C3 and C4 channels. It learned to neglect the blink in all conditions except when the frequency of blinks in the biased condition is very high (90% of trials). On the other hand, for the worst-performing subject, the model is focusing on the blink in all the conditions mainly. The relevancy scores are assigned to regions surrounding the blinks in all the cases. Interestingly, the relevance for the non-blink segment region for this participant increased in the case wherein blinks bias one class the most (90% trials).

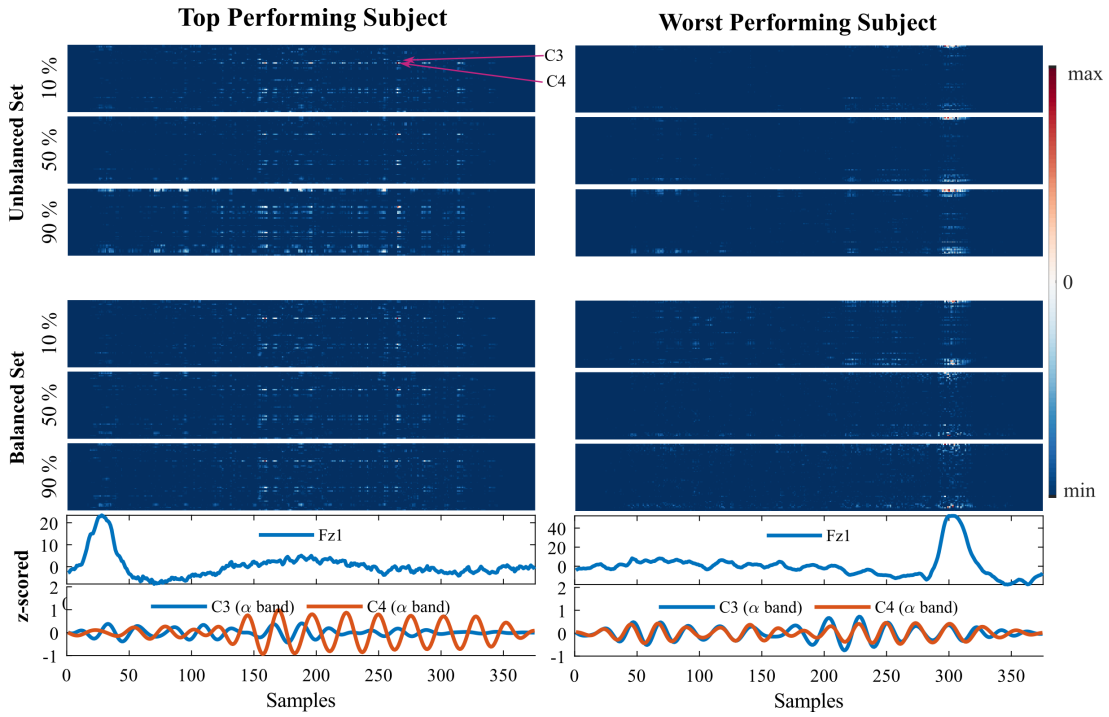

Figure 5.6: DeepLift model explanation showing where the model trained for each condition (one per row) would look to make the correct prediction. Explanations are from the best performing subject (left) and the worst performing subject (right).

5.4 Discussion

Considering that an overwhelming number of studies employing deep learning does not employ artifact handling in their analysis, it was important to evaluate the role the presence of artifact play on the decoding performance. Here, the analysis was limited to eye blinks but the framework could easily be transferred to multiple artifact types. Blinks were chosen because they will be universally present in all studies unless the paradigm involves eye-closed conditions and also due to the relatively high difference in the magnitude of the artifact compared to the signal. CNN can be thought of as a pattern matching framework that tries to find filters that maximally activate them with the global aim of correct prediction of classes of interest. Considering blinks are high amplitude events, it was expected that the frequent presence of blinks

will impact the decoding in opposite directions depending on whether the blinks are biased towards a class or not.

It was also expected that when blinks do bias a particular class, it should significantly improve the decoding in all subjects. Interestingly, that is not the case always, particularly for people in the high decoding category. Evaluating the difference in accuracy as a function of their original decoding on clean EEG 5.5, it was observed that the high-performing subjects are minimally influenced by blinks when they bias the class even 50% of the time. They do end up degrading the decoding in balanced class conditions when the frequency was 50% or 90% of trials. Similarly, in the unbalanced case, the decoding is lower for multiple subjects not in the lower end of the decoding spectrum. This could come from the fact that the other class also has blinks in 10% of the trials. Considering that the presence of blink is not strictly dependent on a particular class, focusing on blinks could lead to many incorrect detections. Particularly for subjects with higher decoding, they would have more consistent class-specific brain rhythms. Even though blinks are higher in one class, the presence of a blink does not necessarily prove it belongs to a particular class as a certain percentage of trials from the other class also contains blinks. Therefore, for these participants, the model might benefit more from learning the consistent brain rhythms instead of focusing on the random presence of blinks.

For some subjects particularly in the lower end of decoding performance, it was observed that including blinks in a high frequency of trials end up improving the decoder even when they do not bias any particular class. The reasoning for that could come from the fact the convolutional neural networks have an implicit bias to learn high-frequency features [156]. Typically the sensorimotor rhythms are in the lower frequency range < 30 Hz (beta or lower). The low-performing subjects, might not have strong signals which can drive the model to learn low-frequency filters to capture these signals. By default, the model might resort to learning high-frequency

filters due to the implicit bias. However, considering blinks are low-frequency signals, when they are introduced in higher frequency, the CNN learns to attend to them in all cases 5.6. By focusing on these blinks, the model would be directed to learn low-frequency filters which might be the reason for the improvement in decoding.

Overall, we systematically compared the influence of the frequency of eye blink artifacts on the CNN decoders, when applied to MI task. We showed that the model decoding is significantly impacted when the frequency of blinks is high. Including explainability approaches can provide some insights into whether the model is influenced by artifacts or not. Currently, no study assesses the influence of artifacts on DL models. We highlight the need for more systematic studies that objectively assess the vulnerability of DL models to artifacts prior to deploying them in an end-to-end manner.

Chapter 6

Conclusion

This dissertation focused on comparing different visualization-based approaches of explainability of deep learning models using both simulated and real EEG. The first chapter motivated the need for explainability in EEG studies. The second chapter used simulated EEG using a realistic head model to compare different model explanation methods to detect distinct EEG features. The sensitivity and robustness of twelve methods were evaluated with respect to detecting EEG features, specifically spectral perturbations, event-related perturbation components, and scalp distribution (spatial). Pitfalls in using some of these methods were identified and DeepLift or LRP methods were found to be the most robust and sensitive measure in general. GradCAM method is a good alternative when looking for spatial patterns but their performance degrades significantly with lower SNR/ model performance. In the third and fourth chapters, the best explainability methods were deployed in two real EEG datasets to evaluate whether they can identify the underlying neural dynamics associated with balance perturbation and motor imagery from single-trial EEG. In the balance perturbation task, it was confirmed that the model was learning from the common PEP components. It was also demonstrated that not handling the artifacts as is done in most deep learning studies currently, will cause the model to be biased by artifacts. Model explanations aligned with prior literature as well as the findings based on the classical signal processing approach. In the motor imagery dataset, the model explanations identified the relevant scalp distribution pertaining to motor imagery aligning with findings from prior studies. Further analysis showed that some participants used strategies/ recruited different channel combinations during the MI. By choosing channel combination in the sensorimotor region as is done in most MI

studies, some of these participants would have degraded decoding performance as the channel montage does not include the channels being recruited the most. Moreover, the analysis showed performance was consistently lower in these subsets of individuals and demonstrated how cluster-specific model explanation could improve the decoding performance, especially for individuals with high decoding scores. All of these approaches suggest the possibility for a novel approach of precision BCI. In the fifth chapter, the impact of eye blinks on the decoding performance was investigated using simulated eye blinks, and it was demonstrated that high frequency of blinks, irrespective of whether they bias either class or not, lead to a significant difference in the decoding. In the case of them biasing the class, including blinks improve the decoding whereas they reduce the decoding performance in the case of balanced condition in general. Interestingly, for high-performing subjects, when blinks are present in a smaller proportion of trials, even when they bias a class, the presence of them negatively impacted the decoding. On the other hand for some of the low-performing subjects having the blinks ended up improving their model performance even when they do not bias any particular class.

All of this comes with different limitations and pitfalls. As discussed earlier, the model explanations were compared based on simulated EEG. Even though synthetic EEG provides more control and allows better characterization, many key features present in real EEG would be missing. However, the dissertation introduces a framework and future studies could include newly identified confounds that are critical. The framework will serve as a baseline for future studies as currently, no study uses a simulation framework to study the influence of deep learning or model explanation on EEG features. The framework could also be used to learn the sensitivity of model architecture to distinct EEG features which is an exciting direction. Similarly, visualization approach is a superficial model explanation method. It is not causal and does not give the full picture. Following up visualization explanation with perturbation approach would be an interesting direction. One approach could be to use

visualization-based model explanation to identify the common channel configuration and then, performing different spectral perturbations on the input for those channels, giving additional insight into the spectral composition of signal in that channel group. One additional limitation is the presence of many complicated characteristics in EEG such as how context, session variability, etc, impact the BCI decoder which is currently not well understood. Evaluating context-independent and dependent features will be extremely beneficial for advancing the BCI research. One possible proxy measure for session invariant decoder/feature stability could be to model the explanation variance over multiple sessions to assess the effect of learning, changes in internal state, and other variables.

In conclusion, this research shows how explainability could be integrated into the existing deep learning studies to support the findings. This research introduces a novel approach in which the clustering method coupled with model explanations could uncover the network configurations for various decoding tasks. Currently, there exists no means to get more context about performance improvement when using deep learning models. The approach proposed here can be used to give further confidence in our model predictions and give insights into whether the model is biased by artifacts or not. This research also demonstrated ways in which model explanation can provide us with valuable information to improve the existing decoders. Using model explanations it is also possible to identify the underlying neural dynamics in a purely data-driven manner. The work presented here will provide guidance and recommendations for researchers who are new to explainability research on EEG. This will further promote the use of an explainability approach to deep learning studies.

References

[1] M. M. Shanechi, "Brain–machine interfaces from motor to mood," *Nature neuroscience*, vol. 22, no. 10, pp. 1554–1564, 2019.

[2] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-computer interfaces in medicine," in *Mayo Clinic Proceedings*, vol. 87, no. 3. Elsevier, 2012, pp. 268–279.

[3] U. Chaudhary, N. Birbaumer, and A. Ramos-Murguialday, "Brain–computer interfaces for communication and rehabilitation," *Nature Reviews Neurology*, vol. 12, no. 9, pp. 513–525, 2016.

[4] D. B. Salisbury, T. D. Parsons, K. R. Monden, Z. Trost, and S. J. Driver, "Brain–computer interface for individuals after spinal cord injury," *Rehabilitation psychology*, vol. 61, no. 4, p. 435, 2016.

[5] E. López-Larraz, A. Sarasola-Sanz, N. Irastorza-Landa, N. Birbaumer, and A. Ramos-Murguialday, "Brain-machine interfaces for rehabilitation in stroke: a review," *NeuroRehabilitation*, vol. 43, no. 1, pp. 77–97, 2018.

[6] P. Ponce, A. Molina, D. C. Balderas, and D. Grammatikou, "Brain computer interfaces for cerebral palsy," *Cerebral Palsy-Challenges for the Future*, 2014.

[7] A. Y. Paek, J. A. Brantley, A. S. Ravindran, K. Nathan, Y. He, D. Eguren, J. G. Cruz-Garza, S. Nakagome, D. S. Wickramasuriya, J. Chang, M. R.-A. Mahfuz, R. Amin, N. Bhagat, and J. Contreras-Vidal, "A roadmap towards standards for neurally controlled end effectors," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 2, pp. 84–90, 2021.

[8] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (eeg) classification tasks: a review," *Journal of neural engineering*, vol. 16, no. 3, p. 031001, 2019.

[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] A. Al-Saegh, S. A. Dawwd, and J. M. Abdul-Jabbar, "Deep learning for motor imagery eeg-based classification: A review," *Biomedical Signal Processing and Control*, vol. 63, p. 102172, 2021.

[11] W. Samek and K.-R. Müller, "Towards explainable artificial intelligence," in *Explainable AI: interpreting, explaining and visualizing deep learning*. Springer, 2019, pp. 5–22.

[12] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *arXiv preprint arXiv:1708.08296*, 2017.

[13] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[14] B. J. Evans and F. A. Pasquale, "Product liability suits for fda-regulated ai/ml software," *The Future of Medical Device Regulation: Innovation and Protection (I. Glenn Cohen, Timo Minssen, W. Nicholson Price II, Christopher Robertson & Carmel Shachar eds., Cambridge University Press, 2021 forthcoming)*, 2020.

[15] K. L. Hudson and F. S. Collins, "The 21st century cures act—a view from the nih," *New England Journal of Medicine*, vol. 376, no. 2, pp. 111–113, 2017.

[16] U. Food and D. Administration, "Artificial intelligence/machine learning (ai/ml)-based software as a medical device (samd) action plan," *US Food Drug Admin., White Oak, MD, USA, Tech. Rep*, vol. 145022, 2021.

[17] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a "right to explanation"," *AI magazine*, vol. 38, no. 3, pp. 50–57, 2017.

[18] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nature communications*, vol. 8, no. 1, pp. 1–8, 2017.

[19] F. Horst, S. Lapuschkin, W. Samek, K.-R. Müller, and W. I. Schöllhorn, "Explaining the unique nature of individual gait patterns with deep learning," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[20] T. McGrath, A. Kapishnikov, N. Tomašev, A. Pearce, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik, "Acquisition of chess knowledge in alphazero," *arXiv preprint arXiv:2111.09259*, 2021.

[21] S. Gaube, H. Suresh, M. Raue, A. Merritt, S. J. Berkowitz, E. Lermer, J. F. Coughlin, J. V. Guttag, E. Colak, and M. Ghassemi, "Do as ai say: susceptibility in deployment of clinical decision-aids," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–8, 2021.

[22] P. Chriskos, C. A. Frantzidis, C. M. Nday, P. T. Gkivogkli, P. D. Bamidis, and C. Kourtidou-Papadeli, "A review on current trends in automatic sleep staging through bio-signal recordings and future challenges," *Sleep Medicine Reviews*, vol. 55, p. 101377, 2021.

[23] A. Y. Paek, J. A. Brantley, B. J. Evans, and J. L. Contreras-Vidal, "Concerns in the blurred divisions between medical and consumer neurotechnology," *IEEE Systems Journal*, vol. 15, no. 2, pp. 3069–3080, 2020.

[24] J. LaRocco, M. D. Le, and D.-G. Paeng, "A systemic review of available low-cost EEG headsets used for drowsiness detection," *Frontiers in neuroinformatics*, vol. 14, 2020.

[25] "Brainco company website," https://brainco.tech, accessed: 2021-12-02.

[26] "BrainCO primary school in china suspends use of brainco brainwave tracking headband," https://qz.com/1742279/a-mind-reading-headband-is-facing-backlash-in-china, accessed: 2021-11-19.

[27] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," *Science*, vol. 366, no. 6464, pp. 447–453, 2019.

[28] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Conference on fairness, accountability and transparency*. PMLR, 2018, pp. 77–91.

[29] N. Xie, G. Ras, M. van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *arXiv preprint arXiv:2004.14545*, 2020.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[31] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[32] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[33] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," *arXiv preprint arXiv:1905.00780*, 2019.

[34] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[35] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3145–3153.

[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[37] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.

[38] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "Layercam: exploring hierarchical class activation maps for localization," *IEEE Transactions on Image Processing*, vol. 30, pp. 5875–5888, 2021.

[39] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that?" *arXiv preprint arXiv:1611.07450*, 2016.

[40] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 24–25.

[41] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2528–2535.

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[44] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1021–1028.

[45] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, "Explaining deep neural networks and beyond: A review of methods and applications," *Proceedings of the IEEE*, vol. 109, no. 3, pp. 247–278, 2021.

[46] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[47] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[48] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[49] M. Bakator and D. Radosav, "Deep learning and medical diagnosis: A review of literature," *Multimodal technologies and interaction*, vol. 2, no. 3, p. 47, 2018.

[50] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre,

G. Van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[51] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, and A. Bridgland, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[52] S. Lapuschkin, "Xai for analyzing and unlearning spurious correlations in imagenet."

[53] C. Buckner, "Understanding adversarial examples requires a theory of artefacts for deep learning," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 731–736, 2020.

[54] W. Ma, Y. Gong, G. Zhou, Y. Liu, L. Zhang, and B. He, "A channel-mixing convolutional neural network for motor imagery eeg decoding and feature visualization," *Biomedical Signal Processing and Control*, vol. 70, p. 103021, 2021.

[55] D. Borra, S. Fantozzi, and E. Magosso, "A lightweight multi-scale convolutional neural network for p300 decoding: Analysis of training strategies and uncovering of network decision," *Frontiers in Human Neuroscience*, vol. 15, 2021.

[56] F. M. Aellen, P. Göktepe-Kavis, S. Apostolopoulos, and A. Tzovara, "Convolutional neural networks for decoding electroencephalography responses and visualizing trial by trial changes in discriminant features," *Journal of neuroscience methods*, vol. 364, p. 109367, 2021.

[57] P. Ortega and A. A. Faisal, "Deep learning multimodal fnirs and eeg signals for bimanual grip force decoding," *Journal of Neural Engineering*, vol. 18, no. 4, p. 0460e6, 2021.

[58] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of eeg sleep stage scoring," in *2017 IEEE 27th international workshop on machine learning for signal processing (MLSP)*. IEEE, 2017, pp. 1–6.

[59] A. Farahat, C. Reichert, C. M. Sweeney-Reed, and H. Hinrichs, "Convolutional neural networks for decoding of covert attention focus and saliency maps for eeg feature visualization," *Journal of neural engineering*, vol. 16, no. 6, p. 066010, 2019.

[60] B. Zang, Y. Lin, Z. Liu, and X. Gao, "A deep learning method for single-trial eeg classification in rsvp task based on spatiotemporal features of erps," *Journal of Neural Engineering*, vol. 18, no. 4, p. 0460c8, 2021.

[61] A. Vahid, M. Mückschel, S. Stober, A.-K. Stock, and C. Beste, "Applying deep learning to single-trial eeg data provides evidence for complementary theories on action control," *Communications biology*, vol. 3, no. 1, pp. 1–11, 2020.

[62] J. Wang, S. Liang, D. He, Y. Wang, Y. Wu, and Y. Zhang, "A sequential graph convolutional network with frequency-domain complex network of eeg signals for epilepsy detection," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2020, pp. 785–792.

[63] X. Jin, J. Tang, X. Kong, Y. Peng, J. Cao, Q. Zhao, and W. Kong, "Ctnn: A convolutional tensor-train neural network for multi-task brainprint recognition," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 103–112, 2020.

[64] A. Petrosyan, M. Sinkin, M. Lebedev, and A. Ossadtchi, "Decoding and interpreting cortical signals with a compact convolutional neural network," *Journal of Neural Engineering*, vol. 18, no. 2, p. 026019, 2021.

[65] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[66] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *Neuroimage*, vol. 87, pp. 96–110, 2014.

[67] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial eeg classification," *Journal of neuroscience methods*, vol. 274, pp. 141–145, 2016.

[68] A. S. Ravindran, M. Cestari, C. Malaya, I. John, G. E. Francisco, C. Layne, and J. L. C. Vidal, "Interpretable deep learning models for single trial prediction of balance loss," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 268–273.

[69] X. Zhang, L. Yao, M. Dong, Z. Liu, Y. Zhang, and Y. Li, "Adversarial representation learning for robust patient-independent epileptic seizure detection," *IEEE journal of biomedical and health informatics*, vol. 24, no. 10, pp. 2852–2859, 2020.

[70] V. Gabeff, T. Teijeiro, M. Zapater, L. Cammoun, S. Rheims, P. Ryvlin, and D. Atienza, "Interpreting deep learning models for epileptic seizure detection on eeg signals," *Artificial Intelligence in Medicine*, vol. 117, p. 102084, 2021.

[71] A. S. Ravindran, A. Mobiny, J. G. Cruz-Garza, A. Paek, A. Kopteva, and J. L. C. Vidal, "Assaying neural activity of children during video game play in public spaces: a deep learning approach," *Journal of neural engineering*, vol. 16, no. 3, p. 036028, 2019.

[72] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human brain mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[73] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Hierarchical internal representation of spectral features in deep convolutional networks trained for eeg decoding," in *2018 6th International Conference on Brain-Computer Interface (BCI)*. IEEE, 2018, pp. 1–6.

[74] R. Mane, N. Robinson, A. P. Vinod, S.-W. Lee, and C. Guan, "A multi-view cnn with novel variance layer for motor imagery brain computer interface," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 2950–2953.

[75] Y. Li, J. Xiang, and T. Kesavadas, "Convolutional correlation analysis for enhancing the performance of ssvep-based brain-computer interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 12, pp. 2681–2690, 2020.

[76] A. H. Thomas, A. Aminifar, and D. Atienza, "Noise-resilient and interpretable epileptic seizure detection," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5.

[77] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un) reliability of saliency methods," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 267–280.

[78] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *arXiv preprint arXiv:1810.03292*, 2018.

[79] H. Barbas and M. Á. García-Cabezas, "Motor cortex layer 4: less is more," *Trends in neurosciences*, vol. 38, no. 5, pp. 259–261, 2015.

[80] M. A. Castro-Alamancos, "The motor cortex: a network tuned to 7-14 hz," *Frontiers in Neural Circuits*, vol. 7, p. 21, 2013.

[81] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[82] L. R. Krol, J. Pawlitzki, F. Lotte, K. Gramann, and T. O. Zander, "Sereega: Simulating event-related eeg activity," *Journal of Neuroscience Methods*, vol. 309, pp. 13–24, 2018.

[83] Y. Huang, L. C. Parra, and S. Haufe, "The new york head—a precise standardized volume conductor model for eeg source localization and tes targeting," *NeuroImage*, vol. 140, pp. 150–162, 2016.

[84] J. C. Mazziotta, A. W. Toga, A. Evans, P. Fox, and J. Lancaster, "A probabilistic atlas of the human brain: theory and rationale for its development," *Neuroimage*, vol. 2, no. 2, pp. 89–101, 1995.

[85] A. Marlin, "Localization of cortical potentials evoked by balance disturbances," Master's thesis, University of Waterloo, 2011.

[86] A. Marlin, G. Mochizuki, W. R. Staines, and W. E. McIlroy, "Localizing evoked

cortical activity associated with balance reactions: does the anterior cingulate play a role?" *Journal of neurophysiology*, vol. 111, no. 12, pp. 2634–2643, 2014.

[87] A. Mierau, T. Hülsdünker, and H. K. Strüder, "Changes in cortical activity associated with adaptive behavior during repeated balance perturbation of unpredictable timing," *Frontiers in Behavioral Neuroscience*, vol. 9, p. 272, 2015.

[88] J. P. Varghese, R. E. McIlroy, and M. Barnett-Cowan, "Perturbation-evoked potentials: Significance and application in balance control research," *Neuroscience & Biobehavioral Reviews*, vol. 83, pp. 267–280, 2017.

[89] S.-S. Yoo, J.-H. Lee, H. O'Leary, L. P. Panych, and F. A. Jolesz, "Neurofeedback fmri-mediated learning and consolidation of regional brain activation during motor imagery," *International journal of imaging systems and technology*, vol. 18, no. 1, pp. 69–78, 2008.

[90] F. Lebon, U. Horn, M. Domin, and M. Lotze, "Motor imagery training: Kinesthetic imagery strategy and inferior parietal f mri activation," *Human brain mapping*, vol. 39, no. 4, pp. 1805–1813, 2018.

[91] O. Mokienko, A. Chervyakov, S. Kulikova, P. Bobrov, L. Chernikova, A. Frolov, and M. Piradov, "Increased motor cortex excitability during motor imagery in brain-computer interface trained subjects," *Frontiers in computational neuroscience*, vol. 7, p. 168, 2013.

[92] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[93] J. Gildenblat and contributors, "Pytorch library for cam methods," https://github.com/jacobgil/pytorch-grad-cam, 2021.

[94] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *The Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

[95] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[96] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through fsim, ssim, mse and psnr—a comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.

[97] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1488–1499, 2011.

[98] L. Arras, A. Osman, and W. Samek, "Ground truth evaluation of neural network explanations with clevr-xai," *arXiv preprint arXiv:2003.07258*, 2020.

[99] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[100] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," *arXiv preprint arXiv:1711.06104*, 2017.

[101] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, "The (un)reliability of saliency methods," 2017.

[102] W. H. Organization, *Falls: fact sheet*, available at https://www.who.int/news-room/fact-sheets/detail/falls.

[103] Y. He, D. Eguren, J. M. Azorín, R. G. Grossman, T. P. Luu, and J. L. Contreras-Vidal, "Brain–machine interfaces for controlling lower-limb powered robotic systems," *Journal of neural engineering*, vol. 15, no. 2, p. 021004, 2018.

[104] D. Pinto-Fernandez, D. Torricelli, M. del Carmen Sanchez-Villamanan, F. Aller, K. Mombaur, R. Conti, N. Vitiello, J. C. Moreno, and J. L. Pons, "Performance evaluation of lower limb exoskeletons: a systematic review," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 7, pp. 1573–1583, 2020.

[105] D. Shi, W. Zhang, W. Zhang, and X. Ding, "A review on lower limb rehabilitation exoskeleton robots," *Chinese Journal of Mechanical Engineering*, vol. 32, no. 1, pp. 1–11, 2019.

[106] Y. W. Hong, Y. King, W. Yeo, C. Ting, Y. Chuah, J. Lee, and E.-T. Chok, "Lower extremity exoskeleton: review and challenges surrounding the technology and its role in rehabilitation of lower limbs," *Australian Journal of Basic and Applied Sciences*, vol. 7, no. 7, pp. 520–524, 2013.

[107] J. L. Contreras-Vidal, N. A. Bhagat, J. Brantley, J. G. Cruz-Garza, Y. He, Q. Manley, S. Nakagome, K. Nathan, S. H. Tan, F. Zhu, and J. L. Pons, "Powered exoskeletons for bipedal locomotion after spinal cord injury," *Journal of neural engineering*, vol. 13, no. 3, p. 031001, 2016.

[108] A. Rodríguez-Fernández, J. Lobo-Prat, and J. M. Font-Llagunes, "Systematic review on wearable lower-limb exoskeletons for gait training in neuromuscular impairments," *Journal of neuroengineering and rehabilitation*, vol. 18, no. 1, pp. 1–21, 2021.

[109] Y. He, D. Eguren, T. P. Luu, and J. L. Contreras-Vidal, "Risk management and regulations for lower limb medical exoskeletons: a review," *Medical devices (Auckland, NZ)*, vol. 10, p. 89, 2017.

[110] C.-H. Wu, H.-F. Mao, J.-S. Hu, T.-Y. Wang, Y.-J. Tsai, and W.-L. Hsu, "The effects of gait training using powered lower limb exoskeleton robot on individuals

with complete spinal cord injury," *Journal of neuroengineering and rehabilitation*, vol. 15, no. 1, pp. 1–10, 2018.

[111] S. Ringhof, I. Patzer, J. Beil, T. Asfour, and T. Stein, "Does a passive unilateral lower limb exoskeleton affect human static and dynamic balance control?" *Frontiers in Sports and Active Living*, vol. 1, p. 22, 2019.

[112] B. Steinhilber, R. Seibt, M. A. Rieger, and T. Luger, "Postural control when using an industrial lower limb exoskeleton: Impact of reaching for a working tool and external perturbation," *Human Factors*, p. 0018720820957466, 2020.

[113] M. Khalili, J. F. Borisoff, and H. M. Van der Loos, "Developing safe fall strategies for lower limb exoskeletons," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 314–319.

[114] V. Monaco, P. Tropea, F. Aprigliano, D. Martelli, A. Parri, M. Cortese, R. Molino-Lova, N. Vitiello, and S. Micera, "An ecologically-controlled exoskeleton can improve balance recovery after slippage," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[115] K. Takakusaki, "Functional neuroanatomy for posture and gait control," *Journal of movement disorders*, vol. 10, no. 1, p. 1, 2017.

[116] J. C. Ditz, A. Schwarz, and G. R. Müller-Putz, "Perturbation-evoked potentials can be classified from single-trial eeg," *Journal of neural engineering*, vol. 17, no. 3, p. 036008, 2020.

[117] D. Tanner, K. Morgan-Short, and S. J. Luck, "How inappropriate high-pass filters can produce artifactual effects and incorrect conclusions in erp studies of language and cognition," *Psychophysiology*, vol. 52, no. 8, pp. 997–1009, 2015.

[118] 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, p. 026013, 2016.

[119] T. R. Mullen, C. A. Kothe, Y. M. Chi, A. Ojeda, T. Kerth, S. Makeig, T.-P. Jung, and G. Cauwenberghs, "Real-time neuroimaging and cognitive monitoring using wearable dry eeg," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2553–2567, 2015.

[120] C.-Y. Chang, S.-H. Hsu, L. Pion-Tonachini, and T.-P. Jung, "Evaluation of artifact subspace reconstruction for automatic eeg artifact removal," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 1242–1245.

[121] A. Delorme and S. Makeig, "Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis," *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004.

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

[123] F. Chollet, "Keras," https://keras.io, 2015.

[124] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[125] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st international conference on distributed computing systems workshops*. IEEE, 2011, pp. 166–171.

[126] M. Breakspear, L. M. Williams, and C. J. Stam, "A novel method for the topographic analysis of neural activity reveals formation and dissolution of 'dynamic cell assemblies'," *Journal of computational neuroscience*, vol. 16, no. 1, pp. 49–68, 2004.

[127] 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.

[128] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[129] A. M. Payne, G. Hajcak, and L. H. Ting, "Dissociation of muscle and cortical response scaling to balance perturbation acceleration," *Journal of neurophysiology*, vol. 121, no. 3, pp. 867–880, 2019.

[130] R. Goel, R. A. Ozdemir, S. Nakagome, J. L. Contreras-Vidal, W. H. Paloski, and P. J. Parikh, "Effects of speed and direction of perturbation on electroencephalographic and balance responses," *Experimental brain research*, vol. 236, no. 7, pp. 2073–2083, 2018.

[131] E. Wittenberg, J. Thompson, C. S. Nam, and J. R. Franz, "Neuroimaging of human balance control: a systematic review," *Frontiers in human neuroscience*, vol. 11, p. 170, 2017.

[132] J. P. Varghese, A. Marlin, K. B. Beyer, W. R. Staines, G. Mochizuki, and W. E. McIlroy, "Frequency characteristics of cortical activity associated with perturbations to upright stability," *Neuroscience letters*, vol. 578, pp. 33–38, 2014.

[133] R. Goel, S. Nakagome, N. Rao, W. H. Paloski, J. L. Contreras-Vidal, and P. J. Parikh, "Fronto-parietal brain areas contribute to the online control of posture during a continuous balance task," *Neuroscience*, vol. 413, pp. 135–153, 2019.

[134] M. Bamdad, H. Zarshenas, and M. A. Auais, "Application of bci systems in neurorehabilitation: a scoping review," *Disability and Rehabilitation: Assistive Technology*, vol. 10, no. 5, pp. 355–364, 2015.

[135] R. A. Ramadan and A. V. Vasilakos, "Brain computer interface: control signals review," *Neurocomputing*, vol. 223, pp. 26–44, 2017.

[136] M. Lotze and U. Halsband, "Motor imagery," *Journal of Physiology-paris*, vol. 99, no. 4-6, pp. 386–395, 2006.

[137] C. Vidaurre and B. Blankertz, "Towards a cure for bci illiteracy," *Brain topography*, vol. 23, no. 2, pp. 194–198, 2010.

[138] M. C. Thompson, "Critiquing the concept of bci illiteracy," *Science and engineering ethics*, vol. 25, no. 4, pp. 1217–1233, 2019.

[139] B. Blankertz, C. Sanelli, S. Halder, E. Hammer, A. Kübler, K.-R. Müller, G. Curio, and T. Dickhaus, "Predicting bci performance to study bci illiteracy," *BMC Neurosci*, vol. 10, no. Suppl 1, p. P84, 2009.

[140] S. C. Kleih and A. Kübler, "Psychological factors influencing brain-computer interface (bci) performance," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 3192–3196.

[141] J. Munzert and K. Zentgraf, "Motor imagery and its implications for understanding the motor system," *Progress in brain research*, vol. 174, pp. 219–229, 2009.

[142] M. D. Golub, M. Y. Byron, and S. M. Chase, "Internal models engaged by brain-computer interface control," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 1327–1330.

[143] B. Z. Allison and C. Neuper, "Could anyone use a bci?" in *Brain-computer interfaces*. Springer, 2010, pp. 35–54.

[144] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set a," *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, vol. 16, pp. 1–6, 2008.

[145] R. Leeb, C. Brunner, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, "Bci competition 2008–graz data set b," *Graz University of Technology, Austria*, pp. 1–6, 2008.

[146] B. Blankertz, K.-R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. R. Millan, M. Schroder, and N. Birbaumer, "The bci competition iii: Validating alternative approaches to actual bci problems," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 153–159, 2006.

[147] M.-H. Lee, O.-Y. Kwon, Y.-J. Kim, H.-K. Kim, Y.-E. Lee, J. Williamson, S. Fazli, and S.-W. Lee, "Eeg dataset and openbmi toolbox for three bci paradigms: an investigation into bci illiteracy," *GigaScience*, vol. 8, no. 5, p. giz002, 2019.

[148] D. Langlois, S. Chartier, and D. Gosselin, "An introduction to independent

component analysis: Infomax and fastica algorithms," *Tutorials in Quantitative Methods for Psychology*, vol. 6, no. 1, pp. 31–38, 2010.

[149] C. M. Stinear, W. D. Byblow, M. Steyvers, O. Levin, and S. P. Swinnen, "Kinesthetic, but not visual, motor imagery modulates corticomotor excitability," *Experimental brain research*, vol. 168, no. 1-2, pp. 157–164, 2006.

[150] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller, "Imagery of motor actions: Differential effects of kinesthetic and visual–motor mode of imagery in single-trial eeg," *Cognitive brain research*, vol. 25, no. 3, pp. 668–677, 2005.

[151] M. Ahn and S. C. Jun, "Performance variation in motor imagery brain–computer interface: a brief review," *Journal of neuroscience methods*, vol. 243, pp. 103–110, 2015.

[152] J. Schwoebel, C. B. Boronat, and H. B. Coslett, "The man who executed "imagined" movements: evidence for dissociable components of the body schema," *Brain and Cognition*, vol. 50, no. 1, pp. 1–16, 2002.

[153] F. Lebon, M. Lotze, C. M. Stinear, and W. D. Byblow, "Task-dependent interaction between parietal and contralateral primary motor cortex during explicit versus implicit motor imagery," *PLoS One*, vol. 7, no. 5, p. e37850, 2012.

[154] B. Yang, K. Duan, C. Fan, C. Hu, and J. Wang, "Automatic ocular artifacts removal in eeg using deep learning," *Biomedical Signal Processing and Control*, vol. 43, pp. 148–158, 2018.

[155] K. Kleifges, N. Bigdely-Shamlo, S. E. Kerick, and K. A. Robbins, "Blinker: Automated extraction of ocular indices from eeg enabling large-scale analysis," *Frontiers in neuroscience*, vol. 11, p. 12, 2017.

[156] J. O. Caro, Y. Ju, R. Pyle, S. Dey, W. Brendel, F. Anselmi, and A. Patel,

"Local convolutions cause an implicit bias towards high frequency adversarial examples," *arXiv preprint arXiv:2006.11440*, 2020.