Deep learning for c-VEP based brain computer interface systems

Author:
Rohit Vijayakumar
9145281

Supervisors:
Dr. Georg Krempl
Prof. dr. Arno Siebes

External Supervisors:
Prof. Peter Desain
Dr. Jordy Thielen

Master's Thesis
Department of Information and Computing Sciences
Utrecht University

July 13, 2022

Acknowledgements

I would like to thank my supervisors Dr. Georg Krempl and Prof. dr. Arno Siebes for their guidance and feedback. I would like to express my sincere gratitude to my external supervisors Prof. Peter Desain and Dr. Jordy Thielen from the Donders Institute for Brain, Cognition and Behaviour for their continuous support, motivation and immense knowledge. Their guidance helped me throughout the time of research and writing of the thesis.

Besides my supervisors, I would also like to thank all the members of the BCI Lab from the Donders Institute especially Dr. Michael Tangermann, Dr. Sara Ahmadi and Pierre Guetschel for the stimulating discussions. I would also like to express my appreciation to Eduardo Santamaría-Vázquez and Víctor Martínez-Cagigal of the BCI Lab from the Biomedical Engineering Group at the University of Valladolid for sharing their knowledge and expertise.

Last but not the least, I would like to thank my parents Mr. and Mrs. Vijayakumar without whom none of this could be possible.

Abstract

Brain-computer interfaces (BCIs) are systems that provide a direct pathway between the electrical activity in the brain and an external computer. Such interfaces enable the control of applications by analyzing brain signals and translating them into the desired command of the user. BCIs have been mainly used to replace or restore abilities to people disabled by neuromuscular diseases. A specific type of BCI is visual evoked potential (VEP) based BCI which uses visual stimuli to evoke a response in the brain. VEP-based BCIs that use non-periodic stimuli are known as code-modulated visual evoked potential (cVEP) based BCI and have been widely used to control visual speller-based applications that allow users to communicate. Several approaches have been used to decode cVEPs from brain signals. Machine learning (ML) algorithms like linear discriminant analysis (LDA) and canonical correlation analysis (CCA) have been widely used in decoding cVEPs. More recently deep neural networks (DNNs) have also been used to decode cVEPs by extracting high-level features from the data. In general for BCIs, subject-to-subject and session-to-session variability is a big challenge. Although certain ML techniques have been proposed to deal with this issue, DNNs have the potential to perform significantly better in this aspect. In this research, the within-subject and leave one subject out (LOSO) performance of DNNs are investigated in terms of accuracy and speed of classification. Improvements in the LOSO performance of the DNN model is further probed using transfer learning to the specific subject. Optimization of the time required for classification is also examined using dynamic stopping methods. Further introspection and visualization of the feature space learned by the model provides an understanding of the spatial and temporal patterns present in cVEP data. The performance of the DNN model is also compared with other prominent approaches (CCA, EEG2Code and EEG-Inception) for decoding c-VEP responses from EEG data.

Contents

List of Figures

List of Tables

Part I

Introduction

Chapter 1

Problem Statement

Rare conditions such as locked-in syndrome affect one percent of people who have a stroke and lose their ability to use certain motor functions as well as their ability to communicate. Late-stage amyotrophic lateral sclerosis (ALS) patients also lose their ability to communicate and control their environments. Brain-computer interfaces (BCIs) [1] have emerged as a possible solution to help such people communicate. One such paradigm is noise-tagging BCI also known as code-modulated visual evoked potential (c-VEP) BCI that uses pseudo-random bit sequences for stimulation [2, 3]. c-VEP based BCI systems have been widely used for speller BCIs [4] where individual symbols are displayed on a screen with each target class of the stimulation sequence being overlayed on the corresponding symbol. The stimuli are a series of flashes with a black frame representing a '0' in the sequence and a white frame representing a '1' in the sequence. When a user overtly attends to one of the symbols, brain responses are evoked corresponding to the flashes in the stimulation sequence that was attended to. A response to a flash in the stimulation sequence is time-locked and occurs around 100ms post-stimulation. Typical c-VEP based BCI systems distinguish between the brain responses to various target classes of stimulation sequences by either decoding responses to individual flashes (on-off) or by decoding the entire stimulation sequence directly from the data. The ability to distinguish between the evoked responses is aided by the design characteristics of stimulation sequences having low auto-correlation and cross-correlation properties. The responses to such stimulation are typically read using electroencephalography (EEG) with electrodes placed on the scalp of the user.

Several machine learning approaches have been used for decoding c-VEPs from brain signals. However, BCI systems typically need to be calibrated for every new user and in some cases even for new sessions with the same user. This cross-session and cross-subject variability of data hinders the performance of a c-VEP based BCI system and its effects are more profound in other VEP domains. Such variability could arise due to several factors such as misplacement of electrodes on the users' scalp, poor signal-to-noise ratio (SNR), low spatial resolutions, etc. Most existing models predict responses to individual events (on-off) present in the stimulation sequence from epoched data (i.e response data corresponding to an individual flash sampled from trial data) [3, 5]. These individual responses are then combined to build a template response for the corresponding stimulation sequence. However, such algorithms have found it challenging to obtain a model that generalizes well to unseen data.

A solution to this problem is a convolutional neural network (CNN) [6] that decodes the entire stimulation sequence directly from the trial data as well as the target class of the corresponding stimulation sequence. The proposed dual-objective CNN model is capable of learning representations of c-VEP based responses in the data irrespective of the subject. Such a model would be able to perform generalized decoding of brain responses to c-VEPs independent of the subject and could be further fine-tuned in real-time using transfer learning [7] to learn subject-specific representations. Although training neural networks is time-consuming, once trained they are extremely fast at making predictions and allow for predicting the stimulation sequence from the brain responses very rapidly. The proposed model is also designed to be able to decode the stimulation sequence from arbitrary durations of data as input with dynamic stopping. This allows for shorter durations of data as input and makes predictions faster taking into account the responsiveness of a BCI system which plays a crucial role in providing a good user experience. Further introspection of the learned weight parameters in such a model would provide insights into the spatial and temporal patterns in c-VEP EEG data. Applying transfer learning to the model would also give an understanding of the extent of fine-tuning required in terms of the amount of calibration data needed for the network to classify unseen subject data accurately.

1.1 Research Questions

- Does the proposed dual-objective CNN model obtain higher performance when tested on within-subject and cross-subject data compared to other models (CCA, EEG2Code and EEG-Inception) in terms of both accuracy and information transfer rate(ITR)

  - Does the performance of the model vary between the target classes of modulated gold codes used as stimulus

  - Can the model allow c-VEP based BCI systems to be used asynchronously by performing non-control state detection (the state at which stimulation sequence is absent)

- How explainable is the model in terms of spatial and temporal patterns obtained from its learned weight parameters on the c-VEP EEG data

- What is the extent of fine-tuning required in terms of the amount of calibration data needed for the network to classify unseen subject data accurately

- Does dynamic stopping based on the confidence score of the model improve classification speed without a significant trade-off in accuracy (i.e. a higher information transfer rate (ITR)

Chapter 2

Background

2.1 Brain-computer interfaces

Brain-computer interfaces (BCI) allow subjects to interact with a computer without using muscular control. This enables motor-disabled people to translate their intentions into application commands helping them regain their communication ability or to control systems required for daily life. Recent work on BCIs have studied expanding its use cases to neurorehabilitation [8], cognitive training [9] and mental state monitoring [10].

There are different kinds of BCI based on the modality of the brain signals being recorded. Completely invasive BCIs [11] place micro-electrodes directly into the cortex having the potential to measure the activity of individual neurons. Non-invasive EEG-based BCIs use sensors that are placed on the scalp to measure electrical potentials generated by the brain. The electrodes measure minute aggregated electrical activities of populations of neurons beneath it. Semi-invasive BCIs usually use electrocorticography (ECoG) where electrodes are placed on the exposed surface of the scalp providing a higher spatial resolution, better signal-to-noise ratio and a wider frequency range than non-invasive BCIs.

Non-invasive BCIs typically use EEG to read the macroscopic electrical activity of the surface layer of the brain. EEG allows to record data from the brain at a high temporal resolution at a low cost and is very portable. However, EEG provides a low spatial resolution and there are a lot of challenges involved with decoding EEG data. The amount of data recorded for a specific experiment is usually restricted to a couple of sessions limiting the amount of data that is acquired. In addition, the signal recorded is weak with a low signal-to-noise ratio as the electrical activity is recorded from the scalp. There is also considerable variation between responses to the same stimuli between different subjects and even for different sessions for the same subject. The data recorded could also be inaccurate due to several factors such as incorrect electrode positioning and incorrect performance of the task by the subject.

2.2 Event related potential (ERP)

BCIs that serve for control and communication applications typically rely on time-locked responses to certain events which are known as event-related potentials (ERP) [12]. ERPs are specific time series patterns recorded from the brain that are time-locked to the presentation of a stimulus. The subject's volitional attention to one of the possible target classes of stimuli allows the BCI to detect the ERP. Based on the (a)periodicity of the stimuli used, there are two kinds of evoked potentials. Visual evoked potentials obtained when using periodic stimuli are known as steady-state visual evoked potentials (SSVEPs) and evoked potentials resulting from non-periodic stimuli being presented to the user are called broad-band visual evoked potentials or code-modulated visual evoked potentials (c-VEPs). SSVEPs have been traditionally used in BCI systems due to their simplicity and speed. They are also known as frequency modulated VEP (f-VEP) as they assign a different stimulation frequency for each specific command and can be decoded in the presence of broadband noise [5]. However, the performance of SSVEP based BCI does not match muscle-based control nor the adequate level of reliability. Broad-band evoked potentials are hypothesized to be more robust to noise due to a broad-band response in the frequency domain and can be decoded even in the presence of narrow-band interference. The non-periodic stimuli are also designed to have minimal auto-correlation and cross-correlation properties hypothesizing that the evoked potentials to the specific stimuli are uncorrelated themselves. However, the responses may not always be uncorrelated even when modelling the brain as a linear system and such effects are more prominent under the assumption that the brain behaves as a non-linear dynamic system.

2.3 c-VEP based BCI system

c-VEPs were first proposed by Sutter [13] as an experimental communication system for severely disabled people. Wei et al. [14] further used c-VEPs as communication applications for subjects suffering from amyotrophic lateral sclerosis (ALS). Using an invasive electrocorticographic system (ECoG), subjects were able to spell 10-12 words per minute. This work was further carried over to EEG-based BCIs by Bin et al. [2] using only one EGG channel (i.e. Oz) showing that c-VEP based BCI systems could attain an accuracy of 91% and an ITR of 92.8 bits/min as compared to the traditional SSVEP based BCI systems that only obtained an 85% accuracy and an ITR of 39.7 bits/min [2]. Martínez-Cagigal et al. [15] gives a detailed overview of c-VEPs and the approaches that have been used for decoding c-VEP responses.

On a typical c-VEP based speller application, responses are recorded to stimuli that are pseudo-random binary noise sequences (PRNS) or codes where a '1' represents a white frame and '0' represents a black frame. Each symbol on the virtual keyboard is associated with a specific stimulation sequence. The virtual keyboard displays target and non-target symbols to the user with associated flash sequences overlayed on each of the corresponding symbols. By attending to a specific symbol, the associated stimulation sequence evokes the corresponding response in the brain which is recorded using EEG. Stimuli can be decoded from such responses thereby classifying the corresponding symbol the user attended to. c-VEP based BCI systems allow for high decoding performances and reduced calibration times.

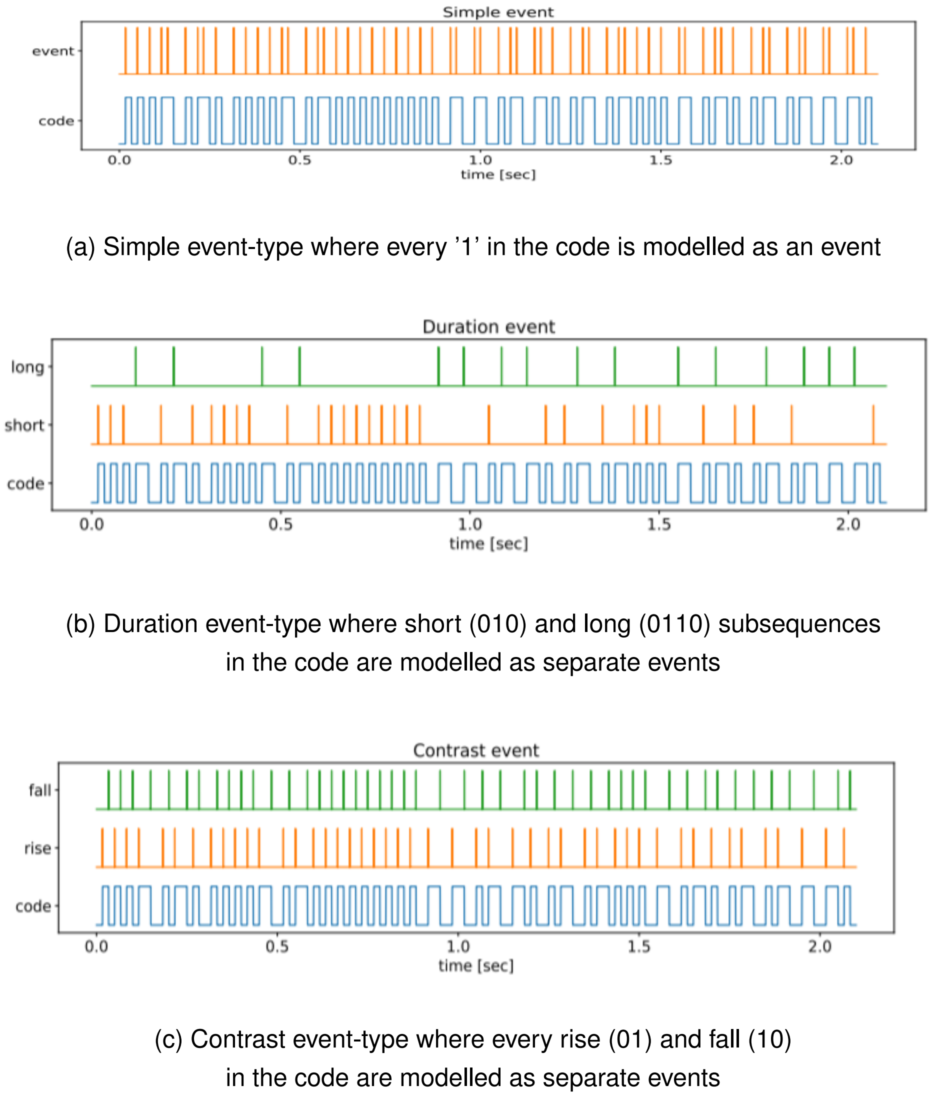
2.4 Code families and event-types

Random binary stimulation sequences need not have optimal correlation properties between them and therefore using a family of codes with suitable auto-correlation and cross-correlation properties is important. However, finding such a family of codes is not trivial [5]. Traditional approaches use pseudo-random binary sequences that have low auto-correlation values for non-zero circular shifts. Each target class is then encoded with the time-delayed version of the original sequence [2]. In c-VEP based BCI systems, m-sequences (maximal length sequences) are often used due to their optimal auto-correlation properties [16]. Such sequences are generated by linear feedback shift registers (LSFR). A 63-bit m-sequence is typically adequate for modulating up to 32 target commands, however longer m-sequences are required for using more targets.

Isaksen et al. [17] compared three code families (m-code, gold-code and barker-code) and concluded that there was no single one that outperformed all the others across subjects. However, for each subject there did appear to be an optimal code that obtained higher accuracy as compared to the other codes. Yasinzai and Ider [18] found that the best results were obtained when using codes with short flashes with enough time between them to ensure that the responses did not overlap. They also found that when repeating flashes of the same length, the responses become weaker with each repetition. Nagel et al. [19] found that optimal results were obtained for codes containing 7 up-down changes for every 15 bits of code.

Modulated gold codes have the special property of being composed of entirely '01' and '10' subsequences. This restricts the stimulation sequence to be only up or down for at most 2 bits at a time. Such stimulation sequences have limited repetitions of the same flash but not the recommended time between flashes to prevent overlap. They are derived from combinations of preferred m-sequence pairs. For m-sequences generated with a polynomial order m, $2^m + 1$ gold codes could be obtained. (e.g. 65 gold codes for a preferred pair of 63-bit m-sequences). For c-VEP based BCI systems, such stimulation sequences are generated prior to the experiment. The gold codes ensure minimal correlation statistics and are further multiplied with a double-bit clock which retains correlation properties but removes low spectrum content.

Another aspect of consideration is the interpretation of events in the stimulation sequence. The definition of events in the stimulation sequence is called the event-type. Considering every '1' (i.e. illuminated display state) in the stimulation sequence as an event is referred to as the 'simple' event-type. Other event-types include the 'duration' event-type where events are defined to be short (010) and long (0110) flashes and the 'contrast' or 'change' event-type where every rise (01) and fall (10) in the stimulation sequence are defined as events. Modelling patterns in the stimulation sequence as multiple events (e.g. long and short flashes) allows for learning more complex non-linear relationships in the c-VEP responses. Figure 2.1 depicts different choices of event-types for a 126-bit gold code (2.1s of data at 60Hz) with optimal correlation properties.



(a) Simple event-type where every '1' in the code is modelled as an event

(b) Duration event-type where short (010) and long (0110) subsequences in the code are modelled as separate events

(c) Contrast event-type where every rise (01) and fall (10) in the code are modelled as separate events

Figure 2.1: Event-types. Adapted from Exploring code families and event-types for cVEP BCIs, by J. Janssen, 2021

Chapter 3

Related Work

In this chapter, an overview of the previous work done in related areas is detailed. These sections will give context to the approaches used in later chapters.

3.1 Traditional approaches

Yasinzai and Ider [18] conducted studies on single-edge VEP responses aimed at predicting the complete c-VEP response to a stimulation sequence using the superposition of responses to individual events in the stimulation sequence. Correlations between the predicted and real ERPs of the corresponding stimulation sequence alone were not able to obtain the desired performance as low correlation coefficients were obtained (e.g. ρ: 0.46). Using a series of constraints that enabled the generation of handcrafted superposition optimized pulse (SOP) sequences, a high correlation between the predicted and real ERPs were obtained (e.g. ρ: 0.79) [15]. They also concluded that although there are non-linear interactions in the c-VEP responses generated to corresponding stimulation sequences, a linear superposition of individual events could obtain accurate predictions for optimized stimulation sequences generated beforehand. Alternative classification methods for decoding c-VEP responses include direct correlation of the c-VEP response with the corresponding stimulation sequence for single trial data [20], support vector machines (SVM) [21–23], one-class SVM (OCSVM) [24–27], linear discriminant analysis [28–30] or naive Bayesian classifiers [31]. However, such models typically reported a loss in performance when the number of target classes was increased.

3.2 Canonical correlation analysis

Recent studies have aimed at improving regression approaches to decoding c-VEP responses from EEG by inferring responses to individual events (e.g. flashes). Learned responses to individual events are further used to predict c-VEP responses to different target classes of stimulation sequences.

Thielen et al. [5, 32] proposed *reconvolution* which generates templates for each user by building up responses to individual events. This approach is based on the linear superposition hypothesis which states that the response evoked by a sequence of events is the linear summation of the evoked responses to individual events. The encoding model learns responses at the level of individual events (i.e. flashes) and is able to generate predictions of responses to full stimulation sequences. There are two steps involved in reconvolution, the estimation-step and the generation-step. In the estimation-step, the full response to a stimulation sequence is decomposed into one or several VEPs (e.g. one for each possible duration of flash). In the generation step, these decomposed responses are combined to generate full template responses. This approach improved on the method proposed by Yasinzai and Ider [18] (limited to single channel data) by utilizing canonical correlation analysis (CCA) that is used along with reconvolution to simultaneously optimize a temporal filter (i.e. transient responses to individual events) and a spatial filter that transforms multiple channels of EEG data into a single channel [33, 34]. CCA optimizes the spatial filter and the temporal filters in one run with the objective of maximizing the correlation between the spatially filtered data and the generated template. Once the CCA model is trained, new unseen data can be classified by first spatial filtering it using the learned spatial filter and then matching it to one of the generated templates. This matching is performed by using Pearson's correlation between the spatially filtered data and each of the individual templates corresponding to the target classes of stimulation sequences and choosing the class with the highest correlation. This method trained with 36 Gold codes achieved a mean online accuracy of 86.0% and a mean information transfer rate (ITR) of 66.4 bits/min in a speller application. Although the duration of the calibration is significantly reduced in c-VEP based BCIs, an adaptive version of the encoding model was proposed to limit the calibration data to none at all or at most a minute [5]. However, completely eliminating the calibration step is not always preferred. For certain user groups, a short calibration phase might provide a covariance estimate that boosts the convergence speed of the system. Results showed that this model was able to reach the same speed and accuracy as the supervised calibrated version. However, without calibration the CCA approach suffers a loss in cross-subject performance.

3.3 Deep learning

Nagel et al. [19, 35, 36] utilized linear ridge regression models based on sliding windows to develop *EEG2Code* and *Code2EEG*. *EEG2Code* takes the c-VEP response as input and the predicts the stimulation sequence that was used to generate it. *Code2EEG* takes a stimulation sequence as input and predicts the associated EEG response. Responses to random stimulation sequences were used to calibrated the models. *EEG2Code* combined with CCA achieved performances around 90% [15] for offline experiments when decoding responses to 1000 different random stimulation sequences.

The method discussed previously assumed linearity (i.e. linear in the model parameters) in the combination of single events to c-VEP response templates. Although this has proven adequate for modelling c-VEP responses, previous research has shown that the brain behaves as a non-linear dynamic system [18, 37]. In order to take the non-linearity of the system into account, Nagel and Spüler [3] combined *EEG2Code* with deep learning. This enabled their model to learn non-linear relationships between the events in the stimulation sequence and c-VEP responses. The model architecture constituted of a convolutional neural network (CNN) and was trained with 384s of data in each trial. On offline experiments using a speller application, the model obtained an accuracy of 98.5% when differentiating between 32 targets [15]. Their model relied on predicting individual flashes from epoched data. However, combining predictions at the flash-level for epoched data from a trial to estimate the target class of the stimulation sequence was not robust enough to obtain high cross-subject performance.

Santamaría-Vázquez et al. [38] proposed a model consisting of a convolutional neural network (CNN) with inception blocks which allows learning temporal patterns at different time scales along with depthwise convolutions (single convolutional filter for each channel) for spatially filtering the multi-channel EEG data. Results of the model trained on a population of subjects on an unseen subject showed accuracies near 90%.

3.4 Transfer learning

One of the major issues that a c-VEP BCI model faces is that a training stage is required for obtaining training data to calibrate its learned weights parameters to adapt it to a specific subject. Ying et al. [39] proposed a Riemannian geometry-based transfer learning algorithm for c-VEP based BCIs that could effectively reduce the calibration time without sacrificing classification accuracy. Santamaría-Vázquez et al. [38] employed transfer learning where a model trained on a population of subjects was used to predict target classes of stimulation sequences for an unseen subject. The model was adapted to each subject for optimizing the subject-specific performance. Although the accuracy was already quite high before adapting the model to the specific subject, fine-tuning the model using small amounts of data from the specific subject enables the model to be more stable by learning subject-specific features. Results of studies that applied transfer learning for SVM and LDA models did not achieve an adequate generalization performance [23, 30].

3.5 Dynamic stopping

Many studies utilized adaptive stopping techniques to dynamically stop the model once it has converged and emit an output [3, 5, 33, 36]. An efficient dynamic stopping approach enables the model to perform at a desired level of accuracy without significantly increasing the time required for making a prediction. These approaches used threshold comparison techniques where the model emitted a target class of stimulation sequence when an optimized measure exceeded a value defined a priori. The measures were obtained from either the correlation coefficients ρ between the c-VEP response data and the learned templates, logistic regression models [40] or transformation into p-values [3, 36]. Other studies used the difference between the first highest and second highest correlation [32]. They applied their algorithm using a sliding window approach allowing their model to make predictions before the end of a cycle. Certain algorithms also applied automatic threshold calibration techniques that enabled their models to be optimized unnoticed by the subject. Dynamic stopping approaches also facilitate the modelling of asynchronous BCI systems to provide self-paced control to the subject. BCI systems are typically synchronous, i.e. they continually produce output predictions based on the EEG activity without a voluntary decision from the subject. A non-control state detection permits the model to monitor the subject's attention and detects whether the subject wants to actively select a command using the c-VEP BCI system. Certain studies provided an asynchronous stage to their BCI system by avoiding command selections that did not surpass a certain threshold.

3.6 Critical reflection

Most of the above studies relied on template-matching algorithms [3.1, 3.2] that use some sort of similarity or distance metric to maximize the correlation between the EEG data input to the model and the learned templates (i.e. the c-VEP responses to stimulation sequences). CCA was most often used to combine information obtained from multiple channels of EEG data. Certain studies [5] also poses the question of whether a c-VEP response can be modeled by the convolution of one basic flash VEP or if the duration of the flash has to be taken into account. Such studies have used basic VEP linear superpositions in generating full templates to recorded c-VEP responses although they state [18] that non-linear interactions contribute a major role in this process. Such models are prone to being not robust without calibration as they are optimized sequentially and due to the non-stationarity of EEG data [41]. Due to low spatial resolutions, poor signal-to-noise-ratio (SNR) and volume conduction in EEG data, the stimulation sequences are not directly reflected in the c-VEP responses and are subject to variability. Robustness to within-subject variability of the c-VEP response data across multiple sessions and cross-subject variability forces the model to require a calibration phase to achieve adequate performance. Predicting individual flashes from epoched response data which are then combined to obtain a prediction for the target stimulation sequence also consumes much more time than predicting the target stimulation sequence directly from arbitrary durations of data.

Deep learning has shown its potential to learn representations that generalize well over unseen data in domains like natural language processing and image classification [38]. CNN architectures for c-VEP based BCI allow for learning non-linear relationships in the EEG data and also optimizes multiple components together instead of doing it sequentially. These networks are designed to optimize multiple spatial and temporal filters in a hierarchical manner. Whereas current models based on correlation predict outputs at the flash level (i.e unique events in the stimulation sequence), deep learning models permit learning temporal structures in the c-VEP response data at the level of trial data. Explainable deep learning models [3] in terms of the spatial and temporal patterns obtained from the learned weight parameters could provide insights into the non-linear generation of c-VEP responses. Deep learning models obtain higher cross-subject performance than traditional approaches in terms of accuracy which could be further optimized using transfer learning where the pre-trained network can be fine-tuned to data from an unseen subject. DNNs allow making predictions of the target stimulation sequence directly from arbitrary lengths of data. Hence the prediction time is much less compared to the standard approaches that rely on predicting responses to individual flashes in the stimulation sequence using epoched data. Assuming such DNNs provide a speed-up compared to standard approaches, incorporating dynamic stopping based on confidence scores of the model could potentially improve the speed of classification and thereby optimize the information transfer rate (ITR) even further.

Part II

Methods

Chapter 4

Data

4.1 8-channel dataset

The first dataset used to evaluate the performance of the model was recorded by Thielen et al. [5]. A total of 30 subjects participated in the experiment. 20 target classes of modulated gold codes [42] of length 126 bits were pre-defined as stimuli. The experiment was based on synchronous control with each trial having a fixed time interval. The EEG data was recorded with 8 sintered Ag/AgCl active electrodes (FpZ, T7, O1, POz, Oz, Iz, O2, T8) at a rate of 512Hz with a Biosemi ActiveTwo amplifier. The codes were displayed on a monitor with a refresh rate of 60Hz and so required 2.1s to present a unique stimulation sequence. Each such sequence was repeated 15 times for a total trial length of 31.5s with an inter-trail time of 1s. The data recorded during the inter-trial time is also used for predicting a non-control state where stimulation is absent. The experiment was conducted using a calculator application with 20 symbols. Each subject completed 5 EEG runs with each run consisting of 20 trials corresponding to each of the 20 target classes. A total of 100 trials were conducted for each participant, with 5 trials for each stimulation sequence making the dataset balanced. For the non-control state, 75 instances of 1s of inter-trial data were collected. Figure 4.2 depicts the recorded EEG channels according to the 10/20 system of placement of electrodes.

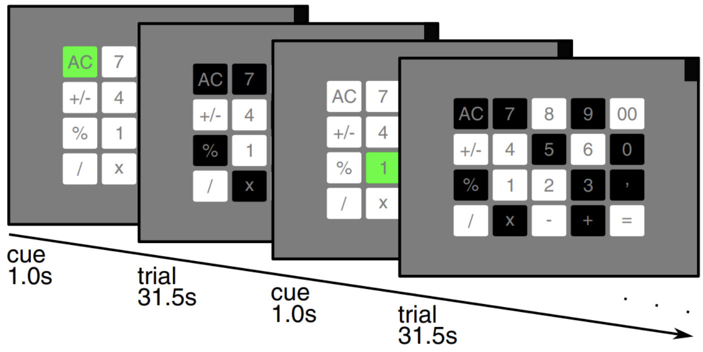


Figure 4.1: Stimuli and experiment where targets were cued with a green color within a 1s inter-trial interval and trials lasted 31.5 s. Adapted from From full calibration to zero training for a code-modulated visual evoked potentials brain computer interface, by J. Thielen et al., 2021

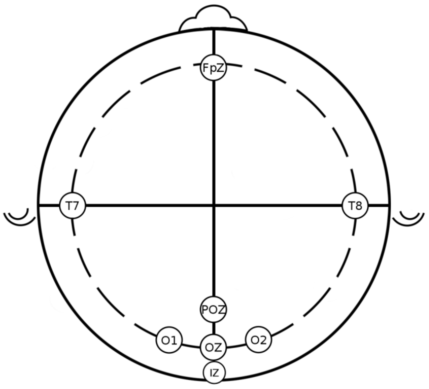


Figure 4.2: 8-channel EEG cap. Adapted from A Deep Learning approach to Noise Tagging, by S. Geurts, 2021

4.2 256-channel dataset

The second dataset used to evaluate the performance of the model was recorded by Ahmadi et al. [43]. 5 subjects participated in the experiment and 36 target classes of modulated gold codes [42] of length 126 bits were pre-defined as stimuli. The subjects were seated at a distance of 60cm from a 17-inch LCD screen with a refresh rate of 60Hz. The EEG data were recorded with a 256-electrode Biosemi cap with gel electrodes at a rate of 360Hz with a Biosemi ActiveTwo amplifier. A stimulation sequence required 2.1s to present to the user with each such sequence being repeated 2 times for a total trial length of 4.2s with an inter-trial time of 0.5s. The stimuli consisted of a flickering matrix-layout keyboard with 36 symbols. Each symbol flickered between black and white frames according to the modulated gold codes. The flickering pattern hence consisted of two types of events, a short flash and a long flash. Each subject completed 3 EEG runs with each run consisting of 36 trials corresponding to each of the 36 target classes. A total of 108 trials were conducted with each participant, 3 trials for each bit-sequence making the dataset balanced. Figure 4.3 depicts the recorded EEG channels according to the Biosemi-256 electrode layout.

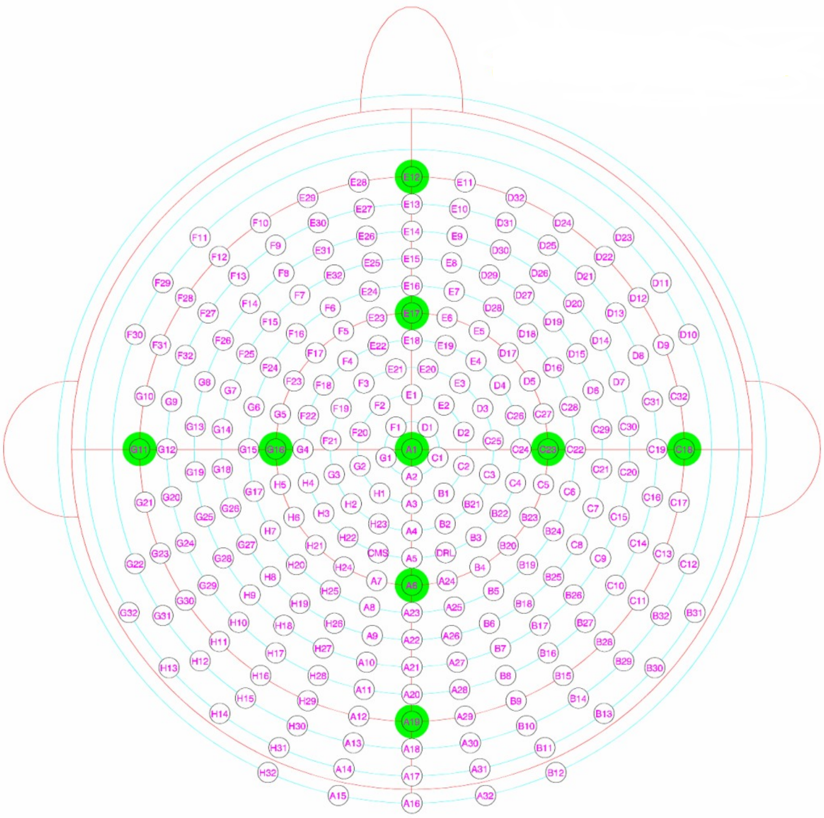


Figure 4.3: 256-channel EEG cap. Adapted from Biosemi, https://www.biosemi.com/headcap

Chapter 5

Implementation

An environment yaml file is used to install all the required dependencies for the code which is coded using Python 3.9.7. The code can be obtained directly from GitHub using a URL to the repository (https://github.com/rohitvk1/Deep-Learning-for-cVEP-based-BCI-systems). The neural networks are implemented using Keras on TensorFlow 2.5.0. The training of the model is done on a PC running Windows 11 with an Intel Core i7-10875H CPU, 16GB of RAM, and an Nvidia GeForce RTX 2070 GPU with 8GB of VRAM.

The following sections describe the methods used for decoding c-VEP responses from EEG data. At first, standard data preprocessing approaches are used (i.e. removing EEG channels with a high standard deviation from the mean, filtering the EEG data to a specific frequency range, resampling the data, augmenting the data with Gaussian noise, and standardizing the data between a specific range). Since full trial data consists of responses to multiple repetitions of the same stimulation sequence, such repetitions are separated during preprocessing. The preprocessed data is further passed to the decoding pipeline where multiple models are compared. The proposed model architecture is a dual-objective CNN that can predict both the target class of the stimulation sequence as well as decode the bits in the stimulation sequence directly from the data. The model uses a masking layer to be able to predict stimulation sequences from an arbitrary duration of input data. This avoids the standard approach of epoching the response data to obtain predictions for individual flashes and further combining them to obtain a target stimulation sequence prediction. The model trained on a population of subjects is further fine-tuned to an unseen subject by using cross-subject transfer learning where the last layers of the model used for classification are retrained to obtain subject-specific weight parameters. Dynamic stopping is also incorporated into the model based on the confidence scores of its predictions to improve the classification speed and thereby the information transfer rate (ITR) even further. The weight parameters of the trained model are also visualized using various algorithms to obtain an explainable model that provides insights into the spatial and temporal patterns in the data. The predictions made by the model are further evaluated using various metrics (e.g. accuracy, f1-score, ITR, etc). Since the model can handle data of arbitrary duration, the performance of the model using these metrics can also be evaluated over time with the corresponding data. The performance obtained by the dual-objective CNN is compared with various models which include CCA, EEG2Code and EEG-Inception.

5.1 Data Preprocessing

The following subsections detail the preprocessing stage of the pipeline. The raw EEG data is preprocessed to improve the signal-to-noise ratio (SNR) of the recorded c-VEP responses.

5.1.1 Removing bad channels

Outliers among the EEG channels recorded are removed by discarding channels that are more than 3 standard deviations away when averaged across trials for each specific subject. A threshold of 3 standard deviations is selected so that only the channels with extreme outliers get rejected. This stage of preprocessing assumes that the data is Gaussian distributed and is not applied to the 8-channel dataset due to the limited number of channels recorded. Figure 5.2 illustrates the channels obtained as outliers in the 256-channel dataset for subject 1 and subject 2 respectively.

5.1.2 Filtering

The data is further preprocessed using a high-pass Butterworth filter of the second order at 2Hz to reject noise at lower frequencies and a low-pass Butterworth filter of the sixth order at 30Hz to reject noise at higher frequencies in the EEG data [5, 44, 45].

5.1.3 Resampling

The data is also downsampled from the recorded frequency to 240Hz being a multiple of 60. This is because the frame rate of the monitor used for stimulation during the experiments was 60Hz. For the CCA model, the data is downsampled to 60Hz to match the size of the structure matrix which is correlated with the input data. The data is also downsampled to 60Hz for the EEG2Code model so that the number of epoched data instances corresponded to the length of the stimulation sequence. [5, 44, 45].

5.1.4 Slicing

During the experiment, a full trial consisted of 31.5s of datapoints for the 8-channel dataset and 4.2s for the 256-channel dataset. Each full trial consisted of repetitions of responses to 2.1s of stimulation. These repetitions were sliced so that c-VEP response to each repetition of the stimulation sequence is considered a separate trial. This allows to learn temporal patterns of importance when classifying c-VEP response data at smaller time scales and increases the number of trials that the models can be trained on. Each sliced trial of the recorded data consisted of 504 datapoints for each channel at 240 Hz (.i.e 2.1s of data). Although the tail of response to the previous repetition leaks into the successive repetition apart from the first instance, it is ignored and all repetitions are treated the same.

For data input to the EEG2code model that performs a flash-level prediction on corresponding durations of response data, the sliced data is further split into epochs of

250 ms of data per window. The number of datapoints in each window was explicitly chosen based on [33] where the transient responses to a flash were shown to be mostly contained in the first 250ms of the c-VEP response. Since the c-VEP response requires more time than that required for presenting a new event, there is an overlap between the epochs. The time required to present a new event is approximately 1/60 seconds (i.e. 17 ms) as the frame rate of the monitor used for stimulation is 60Hz. Therefore, an epoch has 250 - 17 = 237ms overlap with the following epoch.

5.1.5 Standardization

The sliced data is standardized or rescaled for all neural network models so that the variables in the data have the same scale. Each channel in the recorded EEG data is standardized by removing the mean and scaling to unit variance. Standardization allows the models for learning more robust representations which might not be possible if the data does not resemble Gaussian distributed data. Figure 5.1 depicts the data preprocessing performed on the 8-channel EEG data (band-pass filtering, slicing, resampling and standardization) recorded from channel POz of subject 1.

5.1.6 Augmentation

Since the number of repetitions in each trial for the 256-channel dataset is comparatively less than the 8-channel dataset, data augmentation is used to increase the number of trials that the model can be trained on so as to reduce overfitting. Data augmentation is performed by adding Gaussian noise with a mean of 0 and a standard deviation of 1 to the data and concatenating it to the sliced data trials. For the addition of noise to have a consistent effect on the model, it is required to standardize the data prior to augmentation so that the noise has the same smearing effect on the data from multiple trials. If random noise is added before scaling the data, then the data would need to be rescaled again. The noise is added to the data only during the training of the model and not during evaluation.

5.1.7 Training and testing strategy

The data is further split into subsets for training, validation and testing. Specifically 5-fold chronological cross validation is used for the 8-channel dataset and 3-fold chronological cross validation is used for the 256-channel dataset. Each trial in a fold consists of repetitions of the stimulation sequence. Such repetitions are split after generating the folds so that each fold contains all the repetitions of the corresponding trials.

For the 8-channel dataset which consisted of 100 trials for each subject (with 15 repetitions of the stimulation sequence in 1 trial), the data was split into 5 folds with 20 trials in every fold. Each of the 20 trials consisted of 31.5s of data representing the corresponding 20 target classes of stimulation sequences. The non-control state data collected is also split into folds and concatenated with each control state data fold such that the additional target class is as representative as every other target class in each fold. For the 256-channel dataset, which consisted of 108 trials for each subject (with 2 repetitions of the stimulation sequence in 1 trial), the data was split into 3 folds with 36 trials in every fold. Each of the 36 trials consisted of data representing the corresponding 36 target classes of stimulation sequences.

When training the models for obtaining within-subject performance, each fold is left out for testing, whereas the data from the other folds are split into 75% training and 25% validation subsets after shuffling with the same proportion of target stimulation sequences in both subsets. When training the models for obtaining LOSO (i.e. leave one subject out) performance, data from each subject is held out for testing, whereas the data from the other subjects are split into 75% training and 25% validation subsets after shuffling with the same proportion of target stimulation sequences in both subsets. This split is performed over trials so that data being used for training and validation are representative of trial data across all subjects. The test data from each subject that is left out is further split into the same chronological folds used for evaluating the within-subject performance. When training the models for obtaining cross-subject performance, data from each specific subject is used for training and further tested using data from each other subject.

The train data subset is used for training the network to learn the weight parameters required for classifying the data. The validation data subset was used for hyperparameter optimization and to prevent the network from overfitting by observing how well the network generalizes. The testing data subset was used for evaluating the performance of the trained model on unseen data.

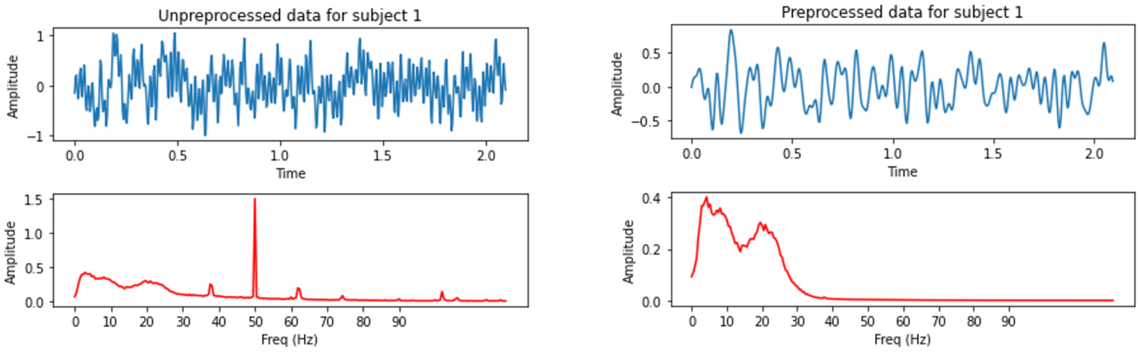


Figure 5.1: Data preprocessing for 8-channel dataset (channel Oz)

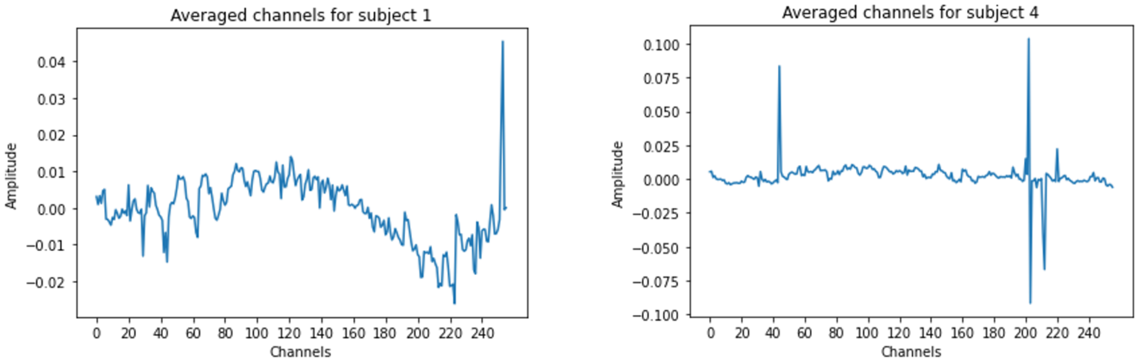


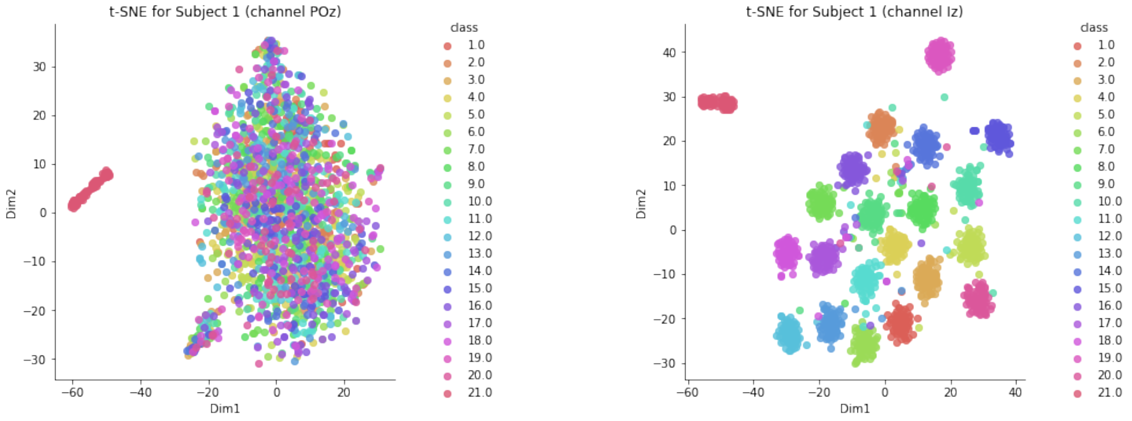
Figure 5.2: Removing bad channels from the 256-channel dataset

5.2 Data Visualization

For visualizing the structure in the recorded high-dimensional EEG data, t-Distributed Stochastic Neighbor Embedding (t-SNE) is used. t-SNE is an unsupervised, non-linear approach used for visualizing and exploring high-dimensional data [46]. t-SNE differs from principal component analysis (PCA) by preserving only small pairwise distances or local similarities. Since PCA is a linear technique that aims to maximize variance, large pairwise dependencies are preserved which could lead to poor visualizations of non-linear structures in the data.

The t-SNE algorithm models the probability distribution of the neighbors around each point using a euclidean distance measure. In the high-dimensional space this is modelled as a Gaussian distribution whereas in the 2-dimensional output space for visualization, it is modelled as a t-distribution. The objective of the algorithm is to find a mapping onto the 2-dimensional output space that minimizes the difference between these distributions over all the points. If the target classes are well-separated by t-SNE, there is a high likelihood that machine learning algorithms and neural networks will be able to learn a mapping from an unseen data point to its ground truth. The main parameter of the t-SNE algorithm is perplexity which corresponds to the number of nearest neighbors when matching the distributions for each point and it defaults to 30. The learning rate defaults to 200 and the gradient calculation algorithm utilizes the Barnes-Hut approximation.

Figure 5.3 depicts the clusters obtained by the t-SNE algorithm in a 2-dimensional output space for channels POz, Iz, Oz and O1 from the 8-channel dataset for subject 1. From this visualization, it is quite evident that the c-VEP response data gets clustered corresponding to the target classes especially for channels closer to the occipital region of the head. The non-control state data recorded (i.e. corresponding to target class 21) gets grouped away from the other clusters corresponding to the target classes in the control state data. For subjects within the 8-channel dataset that the models perform poorly on A.1.1, clusters corresponding to the target classes are not obtained by the t-SNE algorithm.



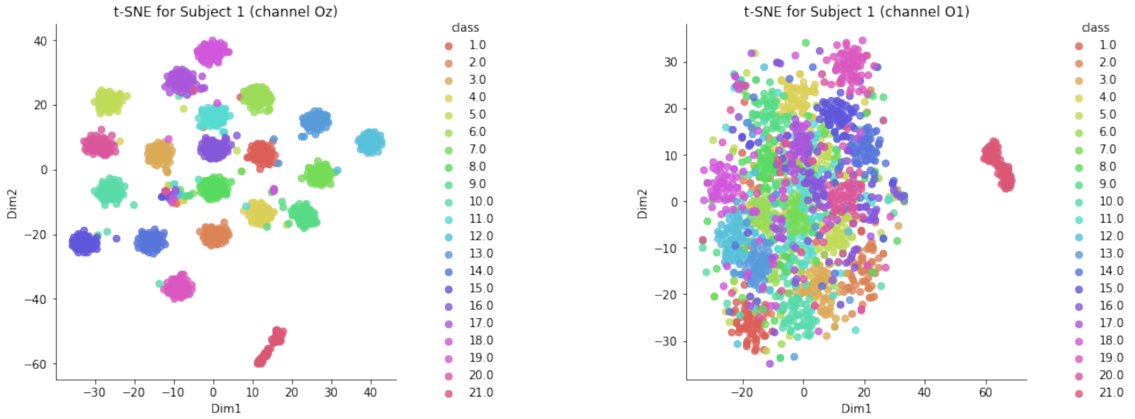


Figure 5.3: t-SNE for 8-channel dataset

5.3 Neural Network

The following subsections detail the neural network model used for feature extraction and command decoding from the preprocessed c-VEP response data.

5.3.1 Model architecture

The model architecture proposed for decoding c-VEP response data is a convolutional neural network (CNN) with two objectives being optimized simultaneously. CNNs have been widely used for EEG processing [3, 47] and have shown exceptional results for synchronous c-VEP based spellers. However only [38] have used deep learning models for decoding the subject's non-control state in a c-VEP based speller. Decoding the non-control state allows the model to work as an asynchronous system that can differentiate between when the subject is actively using the speller (control state) and when the subject is not paying attention to the speller (non-control state).

The architecture of the dual-objective CNN model is depicted in Table 5.1 and is composed of a hierarchy of convolutional blocks. The first block performs spatial filtering on the preprocessed data using 2D convolutions along the spatial axis (i.e. along the channels). The second and third blocks perform temporal filtering on the data with decreasing kernel sizes (.i.e. receptive field sizes) to extract features that are representative of local structure in the data at different temporal scales. These blocks that extract temporal features have kernel sizes that correspond to window sizes of 200ms and 100ms respectively. The stride parameter in each convolution block is used to downsample the data. Lastly, the output block combines the information extracted by the previous blocks into a few high-level features. These high-level features are classified with a softmax output to obtain a target class prediction (.i.e minimizing categorical crossentropy loss) as well as a sigmoid output over the last layer of neurons (.i.e minimizing binary crossentropy loss) whose length corresponds to the length of the stimulation sequence, allowing the model to decode the stimulation sequence directly along with the target class of that stimulation sequence. The model consists of a masking layer at the beginning of the network to allow the classification of input data of varying durations by ignoring regions of the input data (that have been set to 0) in further processing layers. This opens the possibility for obtaining faster predictions by incorporating dynamic stopping into the model. The model is also designed to reduce overfitting by using dropout regularization layers with a dropout rate of 0.25. Additionally, batch normalization and tanh activations were used to improve the performance of the network. The model was trained for 100 iterations along with early stopping which monitored the validation loss and used a batch size of 128. The weights parameters of the model are saved in each iteration where an improvement in validation target class accuracy is obtained.

In contrast to previous work [3, 33] that performed predictions at the flash-level, the proposed model allows for learning more robust features by classifying data at the level of stimulation sequences similar to the EEG-Inception model proposed by Santamaría-Vázquez et al. [38]. However, the proposed model is designed to optimize a dual-objective function in the output block allowing the model to obtain predictions for both the target class of the stimulation sequence as well as decode the stimulation sequence directly from the preprocessed data. This enables the model to obtain more robust predictions with shorter durations of data.

Table 5.1: Dual-objective CNN architecture details

Block	Type	Kernel	Filters	Strides	Output	Connected to
IN	Input	-	-	-	504 x n x 1	M
M	Masking	-	-	-	504 x n x 1	C1
C1	Conv2D	1 x n	8	1	504 x 1 x 8	B1
B1	BatchNorm	-	-	-	504 x 1 x 8	DO1
DO1	Dropout	-	-	-	504 x 1 x 8	C2
C2	Conv2D	48 x 1	8	2	252 x 1 x 8	B2
B2	BatchNorm	-	-	-	252 x 1 x 8	A1
A1	Activation	-	-	-	252 x 1 x 8	DO2
DO2	Dropout	-	-	-	252 x 1 x 8	C3
C3	Conv2D	12 x 1	4	2	126 x 1 x 4	B3
B3	BatchNorm	-	-	-	126 x 1 x 4	A2
A2	Activation	-	-	-	126 x 1 x 4	DO3
DO3	Dropout	-	-	-	126 x 1 x 4	F
F	Flatten	-	-	-	512	D
D	Dense	-	-	-	126	DO4
DO4	Dropout	-	-	-	126	O1, O2
O1	Dense	-	-	-	126	-
O2	Dense	-	-	-	t	-

Column "Type" describes the class used to implement each block in the Keras framework. ‘n’ refers to the number of channels whereas ‘t’ refers to the number of target classes. The model trained on the 8-channel dataset has 85919 parameters of which 85879 are fitted during training whereas the model trained on the 256-channel dataset has 89808 parameters of which 89768 are fitted during training.

5.3.2 Control state detection

For the 8-channel dataset for which the non-control state data is available, the model is designed to predict both the control and non-control states allowing the speller application to work as an asynchronous system. The non-control state data is added as an extra target class along with the classes corresponding to the stimulation sequences for the model to perform classification on. This permits the system to start a new trial without selecting a symbol on the screen when the subject is not overtly attending to the system. When the subject is actively trying to select a symbol on the screen, the model outputs a probabilistic score for each target class of stimulation sequence. The symbol corresponding to the target class of stimulation sequence with the highest confidence score output from the model would be selected by the system.

5.3.3 Feature explainability

Methods for allowing feature explainability in neural networks have become an essential component for analyzing model validation performance. Such methods ensure that the classification performance of the model is driven by the relevant features instead of noise or artifacts in the data. For the proposed dual-objective CNN model, two methods are used for visualizing the learned spatial and temporal patterns respectively. These methods are performed on within-subject data to obtain visualizations for subject-specific spatial and temporal patterns.

For visualizing the spatial patterns learned by the model in its first convolutional layer, the kernel weights in this layer are obtained. Since interpreting the convolutional kernel weights is quite difficult due to the cross-filter map connectivity between layers, the number of spatial filters in the first convolutional layer is restricted to 1 (a special case of model architecture for obtaining insights into the spatial patterns) while ensuring that the decrease in the number of spatial filters does not cause a significant trade-off in accuracy. Once the learned kernel weights that correspond to the number of channels are obtained from the spatial filtering layer, the spatial patterns for any trial data are computed by multiplying the covariance matrix of the data with the learned spatial filter weights. These spatial patterns are representative of the channels in the recorded data that contributed most to the extracted spatial features.

For visualizing the temporal patterns learned by the model in the successive temporal convolutional blocks, Gradient-weighted class activation mapping (Grad-CAM) is used [48]. Grad-CAM uses the gradients of any target class being passed onto the final convolutional layer to generate a coarse localization map highlighting the important regions in the output of that layer for predicting the specific target class. Grad-CAM is used for obtaining target class specific insights. Since Grad-CAM produces an intuitive output, it has been widely used for providing insights into convolutional neural networks especially used for classifying images. To implement Grad-CAM for the c-VEP response data, a model is created that maps the input data to the activations of the last convolutional layer as well as the target class output prediction. The weight parameters during training are loaded into the model. The activation functions in the output block of the model are removed and the gradient of the top predicted class for the input data with respect to the activation (.i.e output feature map) of the last convolutional layer is computed. This produces a vector where each entry is the mean intensity of the gradient over a specific feature map channel. Each channel in the feature map array is further multiplied by the pooled gradients (i.e how important each channel is with regard to the top predicted class) and summed over all the channels to obtain the heatmap class activation. For visualization purposes, this is further normalized to obtain heatmap values between 0 and 1.

5.3.4 Transfer learning

One of the major problems when designing models for decoding c-VEP response data is not being robust enough in terms of classifying different sessions for a certain subject and classifying data from a subject that the model has not seen before. Although the proposed dual-objective CNN model is quite robust in terms of within-subject performance as seen in Section 6, the LOSO (.ie. leave one subject out) classification performance can be further improved for an unseen subject by using transfer learning (.i.e fine-tuning) to calibrate the model to that certain subject.

Transfer learning is used for fine-tuning the learned weight parameters of the dual-objective CNN model trained on a population of subjects to the specific subject that was left out during training (.i.e optimizing the LOSO classification performance of the model). The model is adapted to data from an unseen subject by first freezing all the layers of the model except the output layers (i.e. softmax and sigmoid layers) which are re-trained on validation data of the corresponding subject. This provides insights into the extent of fine-tuning required in terms of the number of additional trials of data required to adapt the model to an unseen subject. Transfer learning allows the model to be adaptive and provides a calibration phase for the model to obtain optimal subject-specific classification performance.

5.3.5 Dynamic stopping

Although the c-VEP response data corresponding to a stimulation sequence has a duration of 2.1s, the model could obtain the correct target class prediction in a shorter amount of time. The masking layer in the dual-objective CNN allows the network to be able to handle arbitrary durations of input data although it is trained on the entire 2.1s window of data in each trial. The information transfer rate (ITR) is an evaluation metric devised for BCI systems that determine the amount of information that is conveyed by a system's output in terms of a trade-off between the time required for classification and the accuracy obtained. The ITR combines the statistics of accuracy and speed of classification as shown in equation 5.1 where P is the classification accuracy, N is the number of target classes and S is the number of trials classified in time T (in minutes). The ITR obtained when evaluating the model over time-steps as shown in Figure 6.5 gives an intuition about how short the duration of input data could be while not sacrificing the classification performance of the model.

$$ITR(\frac{bits}{min}) = (log_2 N + P \times log_2 P + (1-P) \times log_2(\frac{1-P}{N-1})) \times \frac{S}{T} \tag{5.1}$$

To perform dynamic stopping for the model on both the 8-channel and 256-channel datasets, the data is sampled every 100ms. The outlier in the confidence scores over target classes for a particular trial is obtained by estimating the threshold for significance given by the equation 5.2 where k corresponds to the scale required for outlier detection and IQR is the inter-quartile range of the predicted probability scores. The value of k is optimized as a hyperparameter using validation data and is chosen as 1.5. This corresponds with the literature on outlier detection [49] as for most datasets 1.5 controls the sensitivity of the range and thereby the decision rule. A higher value of k would make more outliers to be considered as datapoints whereas a lower value would make some of the datapoints be perceived as outliers. The significant class obtained is also observed every 100ms so that the model outputs a prediction when the same class is emitted 4 times in a row. The probability value or confidence score for that particular class has to also exceed a specific threshold which is optimized using the validation data and is chosen as 0.6. The model incorporated with dynamic stopping is compared with a ceiling performance where only the predicted class has to be the same as the target class 4 times in a row and a static stopping rule. For static stopping, the stopping time is optimized on the validation dataset for each subject thereby having a constant stopping time for all trials within a subject in the test dataset. The dynamic stopping rule based on outlier detection and confidence threshold is also compared with a simpler dynamic stopping rule with just the confidence threshold. Both the dynamic stopping rules emit a target class prediction only after observing 4 consecutive occurrences of the same target class.

$$threshold = IQR \times k + Q3 \tag{5.2}$$

Part III

Results

Chapter 6

Evaluation

6.1 Dual-objective CNN

The following subsections detail the performance of the dual-objective CNN model during training and over various evaluation metrics (accuracy, f1-score, information transfer rate, etc) during testing on both the 8-channel dataset and 256-channel dataset. Insights into the features learned by the model are also obtained by visualizing the spatial and temporal patterns from the learned weight parameters. Furthermore, transfer learning is used to optimize the performance of the model on an unseen subject and dynamic stopping is utilized to improve the speed of classification thereby optimizing the information transfer rate (ITR) of the model. The detailed evaluation metrics of the model for individual subjects in both the datasets are depicted in Tables 6.2, 6.3, 6.4 and 6.5.

6.1.1 Performance during training

Figures 6.1 and 6.2 depicts the performance of the dual-objective CNN in terms of training/validation loss and accuracy on the 8-channel and 256-channel dataset respectively. For the 8-channel dataset, the model generalizes well between training and validation sets for both the within-subject and leave-one-subject-out(LOSO) case as shown in Figure 6.1. However, for the 256-channel dataset the model overfits on the training data especially in the within-subject case as seen in Figure 6.2. Here the model obtains a training accuracy of 100% but attains a validation accuracy of only 40% after training for 100 iterations thereby affecting the generalization performance of the model on unseen data. The reasons behind this overfitting on the 256-channel dataset is further discussed in Subsection 6.1.2.

6.1.2 Accuracy

Figure 6.3 depicts the within-subject and LOSO category accuracy and sequence accuracy averaged across all subjects for both the 8-channel dataset and 256-channel dataset. The category accuracy is evaluated on the softmax output layer of the network and the sequence accuracy is evaluated on the sigmoid output of the network over 126 output neurons (corresponding to the length of the stimulation sequence). Section

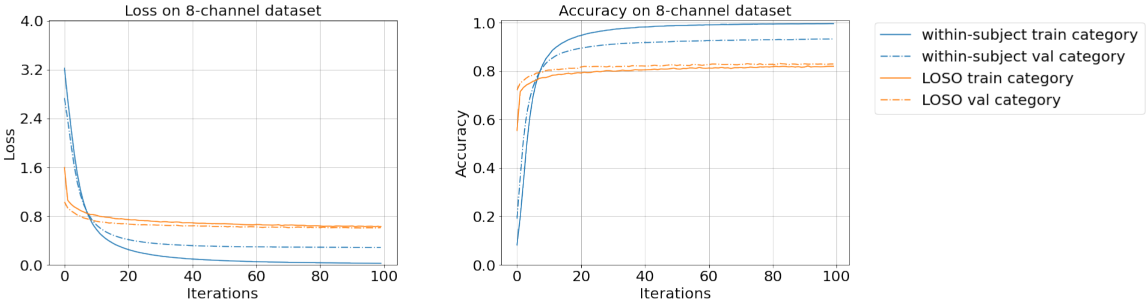


Figure 6.1: Mean training history across folds on 8-channel dataset

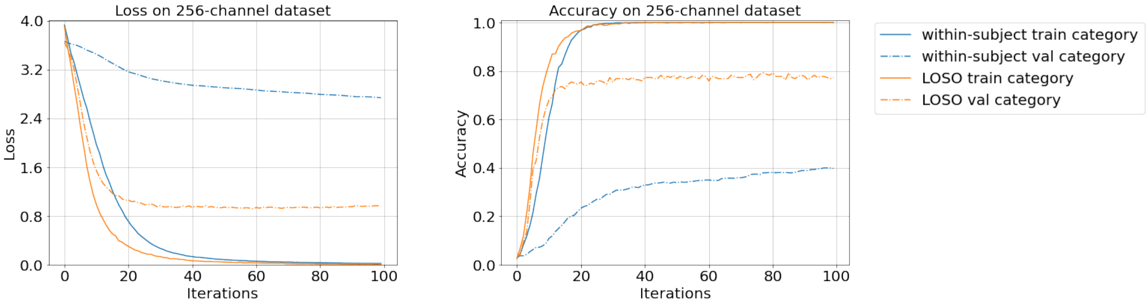


Figure 6.2: Mean training history across folds on 256-channel dataset

A.1.1 illustrates the within-subject and LOSO accuracy for individual subjects in both the datasets. Figures A.9 and A.10 depicts the cross-subject category accuracy obtained on both the datasets. The cross-subject accuracy is indicative of the performance of the model when trained on a particular subject and tested on every other subject in the corresponding dataset.

For the 8-channel dataset, as expected the mean within-subject accuracy is higher than the mean LOSO accuracy as depicted in figure 6.3. This difference in accuracy is observed to be statistically significant and the p-values along with the differences in accuracy on comparing the within-subject performance with the LOSO performance of the model is shown in Table 6.1. However, the mean LOSO performance is higher than the within-subject performance for the 256-channel dataset. The individual subject performance provides an explanation to this observation as only 2 out of the 5 subjects in the 256-channel dataset obtained consistently high within-subject performance. For subjects from the 256-channel dataset that performed poorly, the model overfitted on the corresponding training data as seen in Figure 6.2 and hence the generalization performance on the test data for those subjects were affected. These subjects that exhibited overfitting in the within-subject case was observed consistently across various neural networks (EEG2Code, EEG-Inception) as seen in Figure A.2. However, CCA obtained consistently higher within-subject performance across subjects on the 256-channel dataset. Since the CCA model has far less parameters being optimized as compared to the neural network models, the overfitting of neural network models for the within-subject case in the 256-channel dataset could be attributed to noise in the spatial dimensions that hampers the generalization performance on the validation and test sets. The LOSO performance for these subjects were higher due to the model being able to use optimized spatial filters it learned from a population of subjects within the dataset. The performance of the model in terms of sequence accuracy is comparable to that of category accuracy as both objectives were optimized simultaneously.

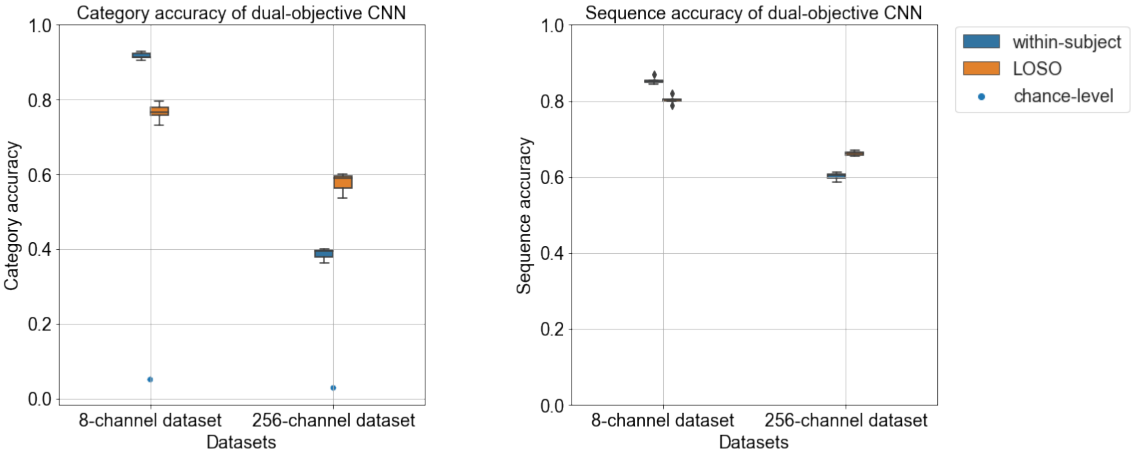


Figure 6.3: Mean category and sequence accuracy across subjects

Datasets	accuracy comparison (%)	t-statistic	α	p-value
8-channel dataset	15.08	13.03	0.05	**0.0**
256-channel dataset	-18.98	-8.2	0.05	**0.001**

Table 6.1: Significance testing on category accuracy between within-subject and LOSO for dual-objective CNN using t-test

6.1.3 F1-score and Information transfer rate(ITR)

The f1-score is the harmonic mean between precision and recall and gives insights into the predictive performance of the model in terms of both the probability of correct detection of a target class and the ability to distinguish between the target classes (true positive rate). Since both the datasets are balanced with the same distribution of corresponding target classes, the f1-score does not provide a lot of additional information to accuracy as seen in Figure 6.4. Section A.1.3 illustrates the within-subject and LOSO f1-score for individual subjects in both the datasets.

The information transfer rate (ITR) is the amount of information transferred in time and is used to measure the performance of communication in terms of both accuracy and speed of classification. The within-subject and LOSO ITR for both the datasets is depicted in Figure 6.4. Section A.1.2 details the within-subject and LOSO ITR for individual subjects in both the datasets.

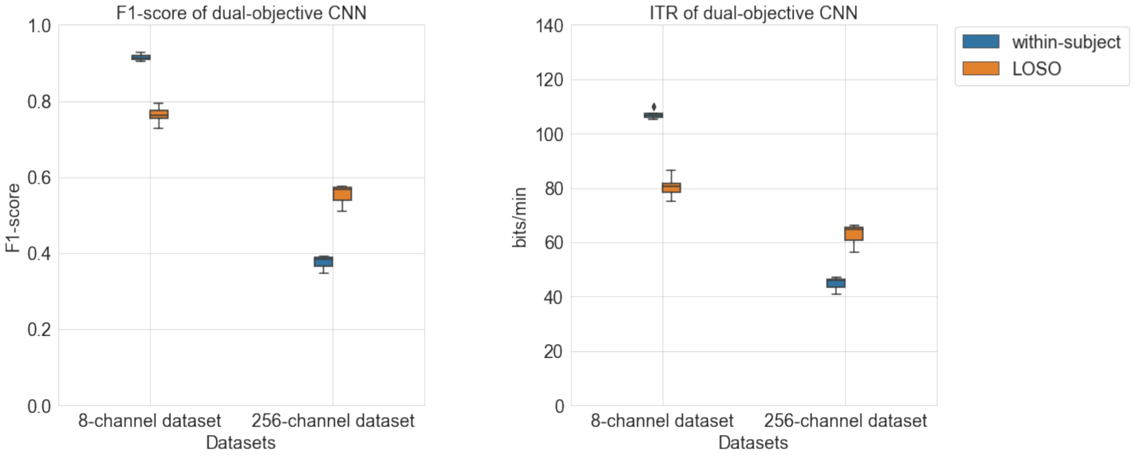


Figure 6.4: Mean f1-score and ITR across subjects

6.1.4 Performance over time-steps

The masking layer in the neural network allows the model to ignore regions within the data enabling it to perform predictions on input data of durations less than 2.1s. This provides insights into the performance of the model at intermediary time-steps and allows the incorporation of dynamic stopping into the model to increase the speed of classification thereby optimizing the information transfer rate (ITR) of the model.

Figure 6.5 depicts the accuracy and information transfer rate (ITR) of the dual-objective CNN model over time. The ITR over time-steps for the model gives insights into when dynamic stopping could be performed so as to optimize the trade-off between classification accuracy and time taken for classification. After first increasing, the ITR decreases as it gets closer to 2.1s of data as the additional data points after the ITR peaks does not give an optimal trade-off between further accuracy improvements and time required for classifying more data points. Section A.1.2 details the within-subject and LOSO ITR over time-steps for individual subjects in both the datasets.

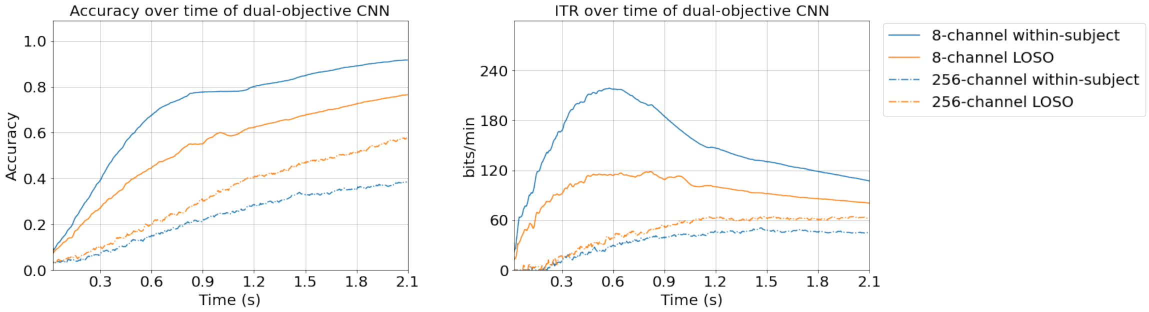


Figure 6.5: Mean Accuracy and ITR over time-steps across subjects

6.1.5 Confusion matrix

A confusion matrix is an N ×N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the trained model. Figure 6.6 depicts the within-subject confusion matrix for the target classes in the 8-channel dataset. The confusion matrix is also normalized for comparing the performance of the model across the ground truths and the predicted target classes. The high values along the diagonal of the confusion matrix shows that the majority of predictions made by the model correspond to the ground truth and further inspection of the confusion matrix shows no outliers in terms of false positives and false negatives among the predicted target classes. Section A.1.4 details the within-subject and LOSO confusion matrix across target classes for both the 8-channel and 256-channel dataset. Since the model optimizes a dual-objective function for predicting the stimulation sequence as well as the target class simultaneously, the within-subject and LOSO flash-level confusion matrices for both the datasets are also depicted in Section A.1.4.

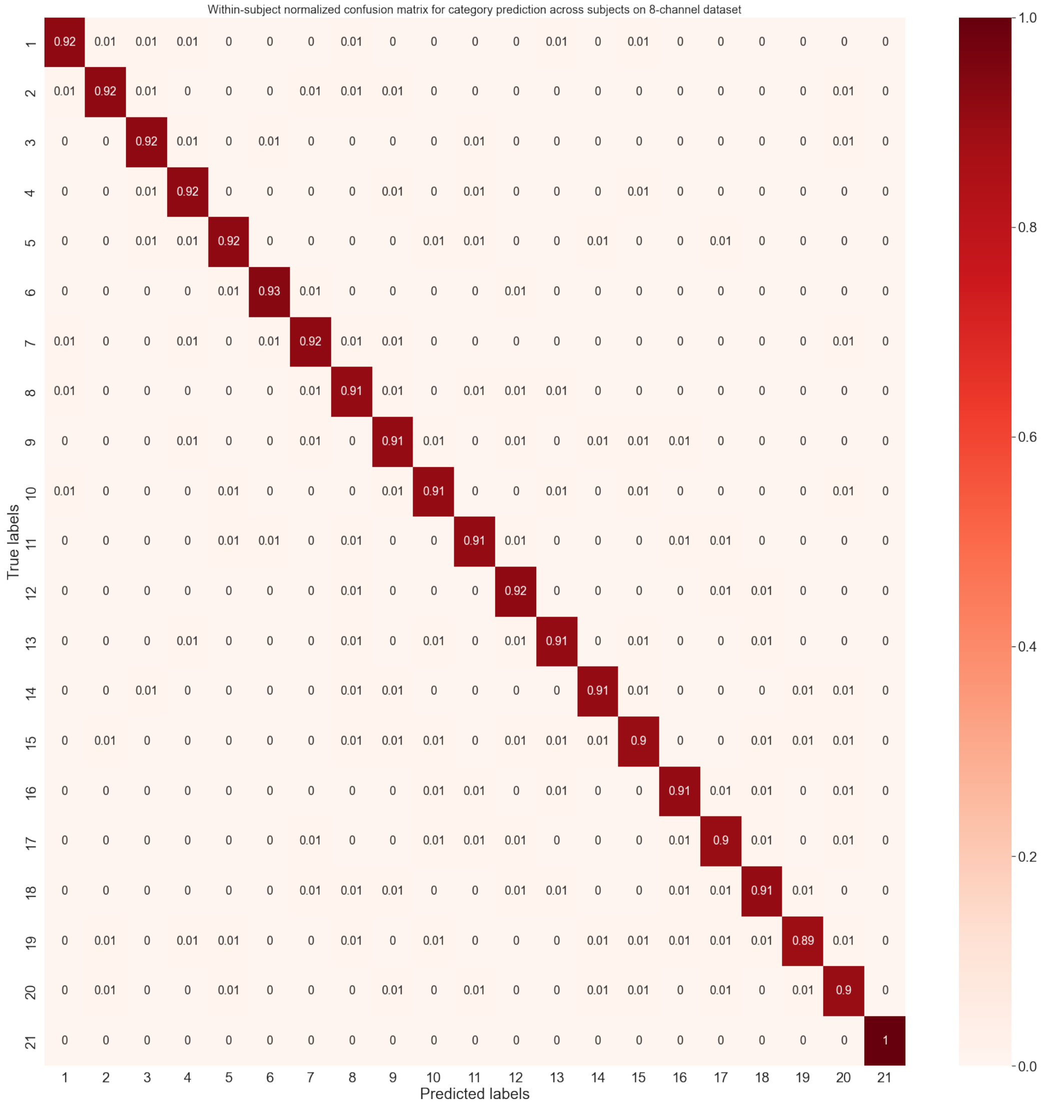


Figure 6.6: Within-subject normalized confusion matrix for category prediction on 8-channel dataset

	category accuracy	sequence accuracy	precision	recall	f1-score	ITR
S01	1.0±0.0	0.83±0.11	1.0±0.0	1.0±0.0	1.0±0.0	124.22±0.0
S02	0.86±0.04	0.81±0.02	0.86±0.04	0.86±0.0	0.85±0.04	90.81±7.36
S03	1.0±0.0	0.92±0.02	1.0±0.0	1.0±0.0	1.0±0.0	124.04±0.88
S04	0.97±0.02	0.9±0.03	0.97±0.03	0.97±0.0	0.97±0.03	117.51±5.65
S05	0.92±0.03	0.84±0.06	0.92±0.04	0.92±0.0	0.91±0.04	104.21±6.38
S06	0.97±0.03	0.88±0.04	0.96±0.04	0.97±0.0	0.96±0.04	116.15±7.43
S07	0.98±0.01	0.92±0.01	0.98±0.01	0.98±0.0	0.97±0.02	118.0±3.81
S08	0.97±0.01	0.87±0.04	0.97±0.01	0.97±0.0	0.97±0.01	116.29±3.97
S09	0.98±0.01	0.89±0.02	0.98±0.01	0.98±0.0	0.98±0.01	118.35±3.78
S10	0.74±0.05	0.75±0.02	0.75±0.05	0.74±0.0	0.74±0.05	69.97±8.09
S11	0.86±0.05	0.81±0.03	0.87±0.05	0.86±0.0	0.86±0.06	92.44±11.49
S12	0.28±0.17	0.6±0.05	0.29±0.17	0.28±0.0	0.28±0.16	15.2±15.62
S13	0.84±0.04	0.8±0.02	0.85±0.04	0.84±0.0	0.84±0.04	87.72±7.29
S14	0.98±0.01	0.89±0.02	0.98±0.01	0.98±0.0	0.98±0.01	118.54±3.89
S15	0.98±0.01	0.93±0.03	0.98±0.01	0.98±0.0	0.98±0.01	120.15±3.31
S16	0.98±0.02	0.93±0.02	0.98±0.02	0.98±0.0	0.98±0.02	120.14±5.05
S17	0.99±0.01	0.89±0.08	0.99±0.01	0.99±0.0	0.99±0.01	123.5±2.07
S18	0.99±0.01	0.95±0.01	0.99±0.01	0.99±0.0	0.99±0.01	122.69±2.75
S19	0.87±0.07	0.82±0.03	0.87±0.08	0.87±0.0	0.86±0.08	93.97±13.84
S20	0.88±0.05	0.82±0.02	0.89±0.05	0.88±0.0	0.88±0.05	95.93±10.41
S21	0.99±0.01	0.91±0.06	0.99±0.01	0.99±0.0	0.99±0.01	123.3±2.52
S22	1.0±0.0	0.77±0.04	1.0±0.0	1.0±0.0	1.0±0.0	125.49±0.0
S23	0.89±0.04	0.81±0.01	0.9±0.04	0.89±0.0	0.89±0.04	98.87±7.44
S24	0.97±0.02	0.89±0.05	0.98±0.02	0.97±0.0	0.97±0.02	117.31±5.36
S25	0.92±0.04	0.88±0.02	0.92±0.03	0.92±0.0	0.91±0.04	103.48±8.0
S26	0.88±0.06	0.81±0.02	0.88±0.06	0.88±0.0	0.87±0.06	95.41±11.79
S27	1.0±0.0	0.91±0.05	1.0±0.0	1.0±0.0	1.0±0.0	124.73±0.62
S28	0.98±0.01	0.89±0.02	0.98±0.01	0.98±0.0	0.98±0.01	118.34±3.46
S29	0.99±0.01	0.86±0.04	1.0±0.01	0.99±0.0	0.99±0.01	123.7±1.97
S30	0.85±0.03	0.81±0.01	0.86±0.03	0.85±0.0	0.85±0.03	90.59±5.66
mean	0.92±0.01	0.85±0.01	0.92±0.01	0.92±0.0	0.92±0.01	107.03±1.69

Table 6.2: Within-subject results of dual-objective CNN on 8-channel dataset

	category accuracy	sequence accuracy	precision	recall	f1-score	ITR
S01	0.81±0.01	0.69±0.01	0.83±0.03	0.81±0.0	0.8±0.01	99.89±2.69
S02	0.13±0.05	0.55±0.02	0.11±0.06	0.12±0.0	0.11±0.05	3.12±4.41
S03	0.03±0.02	0.5±0.0	0.02±0.01	0.03±0.0	0.03±0.01	0.0±0.0
S04	0.9±0.06	0.74±0.03	0.92±0.06	0.9±0.0	0.89±0.06	119.98±13.6
S05	0.06±0.02	0.52±0.01	0.04±0.02	0.06±0.0	0.05±0.02	0.0±0.0
mean	0.39±0.02	0.6±0.01	0.38±0.02	0.39±0.0	0.37±0.02	44.6±2.81

Table 6.3: Within-subject results of dual-objective CNN on 256-channel dataset

	category accuracy	**sequence accuracy**	**precision**	**recall**	**f1-score**	**ITR**
S01	1.0±0.0	0.95±0.01	1.0±0.0	1.0±0.0	1.0±0.0	124.07±1.16
S02	0.58±0.07	0.71±0.02	0.6±0.06	0.58±0.0	0.58±0.07	46.63±9.16
S03	0.89±0.03	0.83±0.01	0.9±0.03	0.89±0.0	0.89±0.03	98.77±5.86
S04	0.62±0.23	0.73±0.07	0.63±0.23	0.62±0.0	0.62±0.23	56.29±32.16
S05	0.84±0.06	0.82±0.03	0.85±0.06	0.84±0.0	0.84±0.06	88.76±11.21
S06	0.94±0.03	0.87±0.02	0.93±0.04	0.94±0.0	0.94±0.03	109.66±6.12
S07	0.88±0.03	0.83±0.01	0.88±0.03	0.88±0.0	0.88±0.03	95.18±5.8
S08	0.8±0.19	0.82±0.08	0.8±0.19	0.8±0.0	0.8±0.19	84.6±33.77
S09	0.93±0.02	0.85±0.01	0.93±0.02	0.93±0.0	0.93±0.02	106.14±4.95
S10	0.35±0.06	0.65±0.02	0.37±0.08	0.35±0.0	0.35±0.06	18.96±6.58
S11	0.8±0.07	0.8±0.03	0.81±0.07	0.8±0.0	0.8±0.07	81.62±12.62
S12	0.42±0.33	0.66±0.11	0.42±0.33	0.42±0.0	0.42±0.33	35.46±39.38
S13	0.59±0.06	0.72±0.02	0.6±0.06	0.59±0.0	0.58±0.06	46.68±7.57
S14	0.64±0.09	0.74±0.03	0.66±0.09	0.64±0.0	0.64±0.09	55.39±13.64
S15	0.9±0.03	0.83±0.02	0.91±0.03	0.9±0.0	0.9±0.03	99.82±7.45
S16	0.88±0.05	0.81±0.02	0.88±0.05	0.88±0.0	0.88±0.05	95.74±9.73
S17	0.97±0.01	0.9±0.01	0.97±0.01	0.97±0.0	0.97±0.01	115.2±3.22
S18	0.93±0.05	0.87±0.04	0.93±0.05	0.93±0.0	0.93±0.05	106.83±12.25
S19	0.26±0.02	0.62±0.01	0.26±0.02	0.26±0.0	0.25±0.02	10.38±2.01
S20	0.75±0.05	0.77±0.02	0.76±0.06	0.75±0.0	0.75±0.05	72.59±8.92
S21	1.0±0.0	0.97±0.01	1.0±0.0	1.0±0.0	1.0±0.0	124.26±0.75
S22	1.0±0.0	0.96±0.01	1.0±0.0	1.0±0.0	1.0±0.0	125.24±0.51
S23	0.44±0.08	0.67±0.02	0.45±0.08	0.44±0.0	0.43±0.08	28.43±8.78
S24	0.66±0.09	0.73±0.03	0.66±0.09	0.66±0.0	0.65±0.09	57.27±13.24
S25	0.73±0.06	0.77±0.01	0.74±0.05	0.73±0.0	0.72±0.06	67.73±9.16
S26	0.54±0.07	0.71±0.02	0.55±0.07	0.54±0.0	0.54±0.07	40.76±8.5
S27	0.99±0.01	0.93±0.01	0.99±0.01	0.99±0.0	0.99±0.01	122.42±2.03
S28	0.92±0.04	0.85±0.02	0.93±0.04	0.92±0.0	0.92±0.04	105.38±8.43
S29	0.99±0.0	0.94±0.0	1.0±0.0	0.99±0.0	0.99±0.0	123.6±1.24
S30	0.74±0.02	0.77±0.0	0.75±0.02	0.74±0.0	0.73±0.02	70.06±2.86
mean	0.77±0.02	0.8±0.01	0.77±0.02	0.77±0.0	0.76±0.02	80.46±3.76

Table 6.4: LOSO results of dual-objective CNN on 8-channel dataset

	category accuracy	**sequence accuracy**	**precision**	**recall**	**f1-score**	**ITR**
S01	0.7±0.05	0.69±0.01	0.71±0.05	0.7±0.0	0.68±0.06	79.46±8.5
S02	0.75±0.03	0.69±0.0	0.76±0.02	0.75±0.0	0.73±0.03	88.9±6.23
S03	0.25±0.01	0.58±0.0	0.23±0.02	0.25±0.0	0.22±0.01	14.16±0.65
S04	0.83±0.05	0.73±0.01	0.86±0.05	0.83±0.0	0.82±0.05	104.05±9.82
S05	0.34±0.09	0.62±0.02	0.32±0.08	0.34±0.0	0.31±0.08	25.55±10.1
mean	0.57±0.03	0.66±0.01	0.58±0.03	0.58±0.0	0.55±0.03	62.42±4.27

Table 6.5: LOSO results of dual-objective CNN on 256-channel dataset

6.1.6 Feature explainability

The explainability of the dual-objective CNN model in terms of spatial and temporal patterns are visualized for the within-subject case to gain insights into the subject-specific spatial and temporal patterns obtained using the learned weight parameters of the model.

6.1.6.1 Spatial patterns

The kernels learned by the model to perform spatial filtering and the corresponding spatial patterns obtained by multiplying the learned kernel weights with the covariance matrix of the test data are visualized in Figure 6.7 and 6.8. The size of the learned kernels correspond to the number of channels in each dataset so as to separate the spatial and temporal feature extraction in the neural network. This facilitates intuitive visualizations of both the spatial and temporal patterns within the data. Since only the magnitude of the kernel weights carry information whereas the sign of the weights do not provide any additional information, such signs are corrected for when visualizing the spatial patterns. The visualized spatial patterns provides insights into the channels that were most informative for the model in classifying the data.

From Figures 6.7 and 6.8, as expected the spatial patterns are concentrated near the occipital region of the head where the primary visual cortex lies that processes visual information which is relayed by the retinas.

6.1.6.2 Temporal patterns

Figure 6.9 and 6.10 illustrates the visualizations obtained by the Grad-CAM algorithm. These heatmaps provide insights into the regions of temporal importance within the data thereby revealing temporal patterns that were most informative for the model in classifying the data.

From Figure 6.9, it is evident that the model gives importance to the first and last regions of the 2.1 window, whereas it does not learn any temporal information from the region in between 0.8 and 1.1s. Therefore for the 8-channel dataset, dynamic stopping algorithms would optimally stop around 0.9s for each trial. This gap in temporal information in the within-subject case for the 8-channel dataset is also evident in the performance of the dual-objective CNN over time-steps for individual subjects as shown in Figures A.5 and A.14. The same pattern is also observed in the within-subject performance over time-steps for various models (CCA, EEG2Code and EEG-Inception) as well which suggests that this gap in temporal information is not model specific but inherent to the 8-channel dataset. For the 256-channel dataset, Figure 6.10 shows regions of temporal importance intermittently along the 2.1s trial and no such prolonged gap in temporal information is observed.

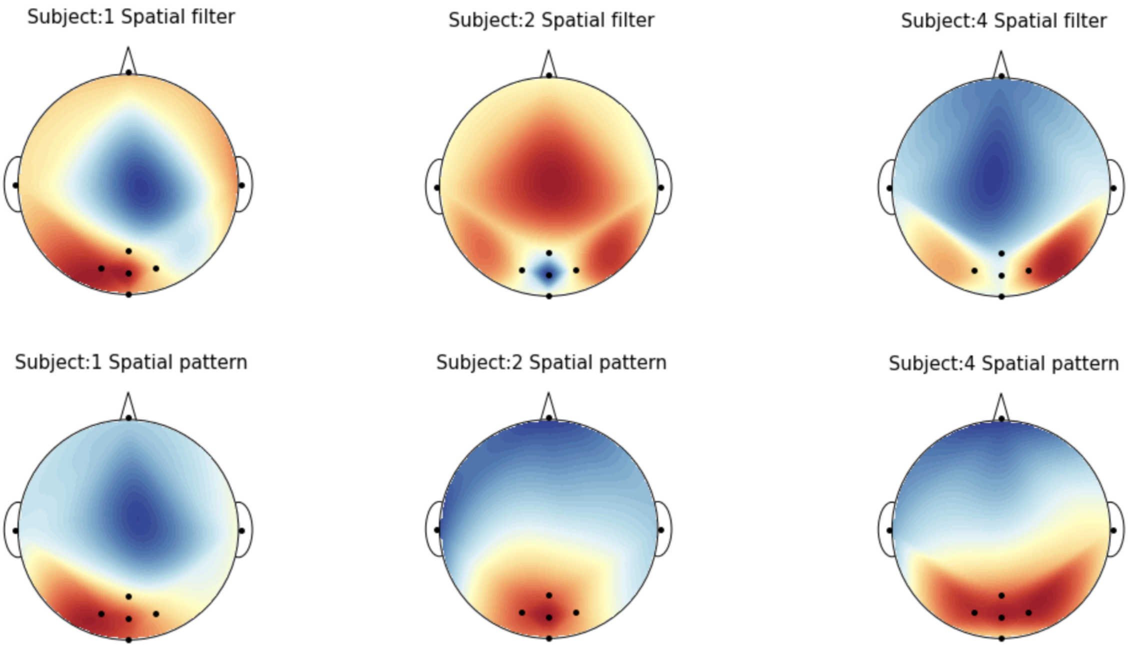


Figure 6.7: Spatial filters and patterns for 8-channel dataset

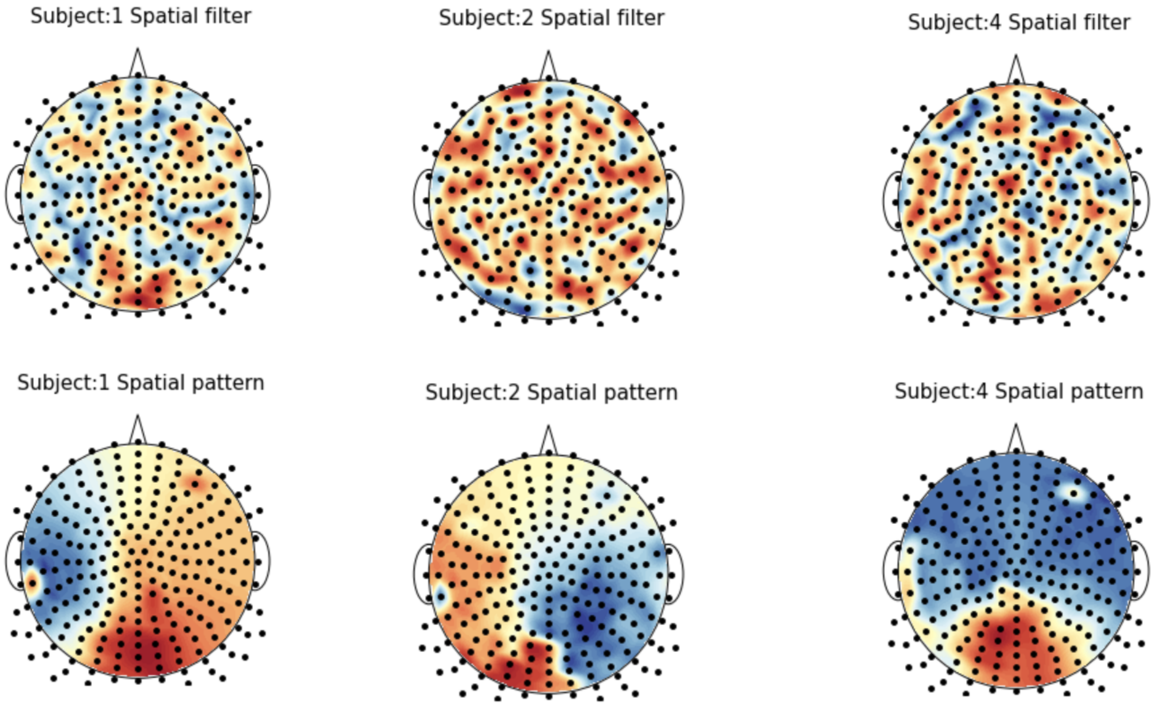


Figure 6.8: Spatial filters and patterns for 256-channel dataset

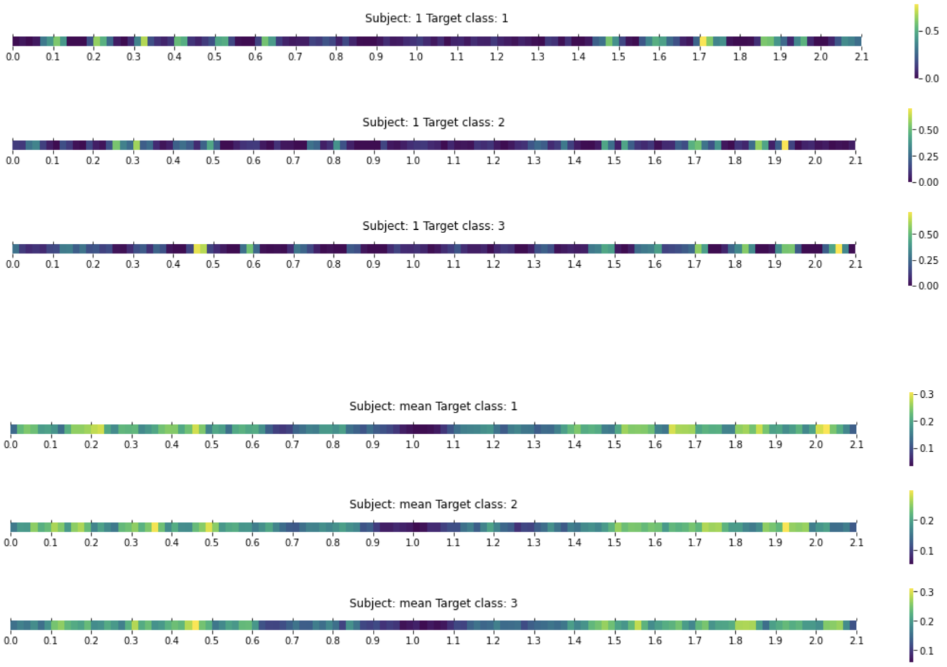


Figure 6.9: Temporal patterns for 8-channel dataset

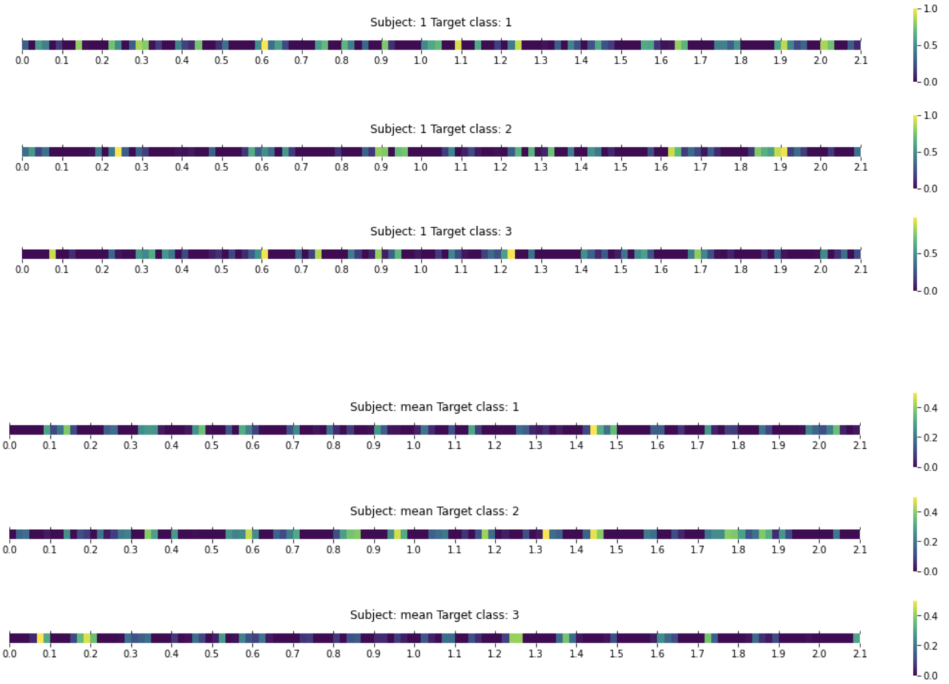


Figure 6.10: Temporal patterns for 256-channel dataset

6.1.7 Transfer learning

Transfer learning allows for fine-tuning the learned weight parameters of the dual-objective CNN model trained on a population of subjects to a specific unseen subject. Figures 6.11 and 6.12 depicts the extent of fine-tuning required in terms of the number of additional trials of data required to adapt the LOSO model to each of the unseen subjects.

For both the datasets, the model required around 20 additional trials (each of duration 2.1s) to adapt its weight parameters for optimizing subject-specific performance. For the 256-channel dataset, Figure 6.12 shows that there is a drop in accuracy in the first few additional trials as the weights in the output layer of the network are re-trained whereas it is not observed consistently across subjects for the 8-channel dataset. This could be due to the model being more generalizable when trained on a large population of subjects in the 8-channel dataset whereas it does not optimally converge for the small population of subjects in the 256-channel dataset requiring for more variation in the output layers of the model during re-training on an unseen subject. For optimizing the LOSO performance of the model on an unseen subject to be closer to the within-subject performance, the spatial filters and temporal filters of the model have to be re-trained. However, this requires a lot of additional trials of data for the model to converge and is not feasible for transfer learning with a limited duration of additional data.

6.1.8 Dynamic stopping

Dynamic stopping permits the dual-objective CNN model to emit target class predictions at much shorter durations than the input data duration of 2.1s. Figure 6.5 depicts the duration of LOSO data required required to obtain the corresponding accuracy levels. As the ITR peaks at around 80% accuracy, an intuitive duration that could be selected for early stopping is 0.9s.

Figure 6.13 illustrates the performance metrics (accuracy, time and ITR) for various early stopping approaches across subjects in both the datasets. The base approach corresponds to using no early stopping and predicting on the entire input duration of 2.1s of data. Although the accuracy is highest for the base case as it has access to more data, it does require the entire 2.1s resulting in the lowest information transfer rate (ITR). The ceiling approach indicates the highest performance that could be achieved by the static and dynamic stopping methods. On comparing the performance between the static stopping and dynamic stopping approaches, it is observed that the dynamic stopping approaches obtain higher accuracies than static stopping. Since the time required for obtaining those accuracies when comparing the dynamic stopping approaches to static stopping are lower for the 8-channel dataset and comparable for the 256-channel dataset, the ITR for the dynamic stopping approaches are higher as compared to static stopping. Both the dynamic stopping methods based on confidence thresholding and outlier detection respectively shows similar performance.

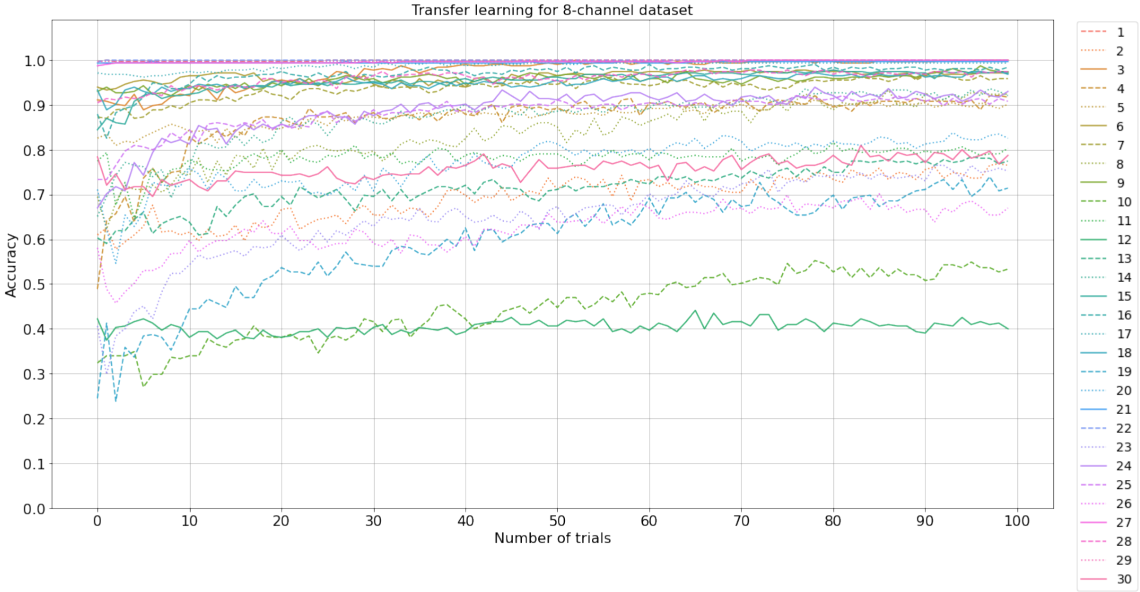


Figure 6.11: Transfer learning for 8-channel dataset

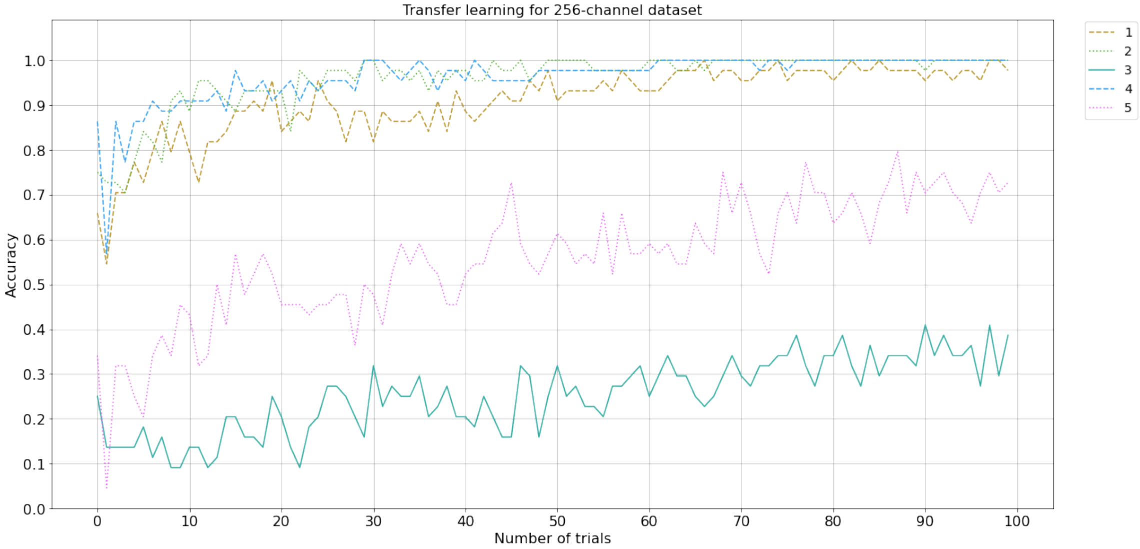


Figure 6.12: Transfer learning for 256-channel dataset

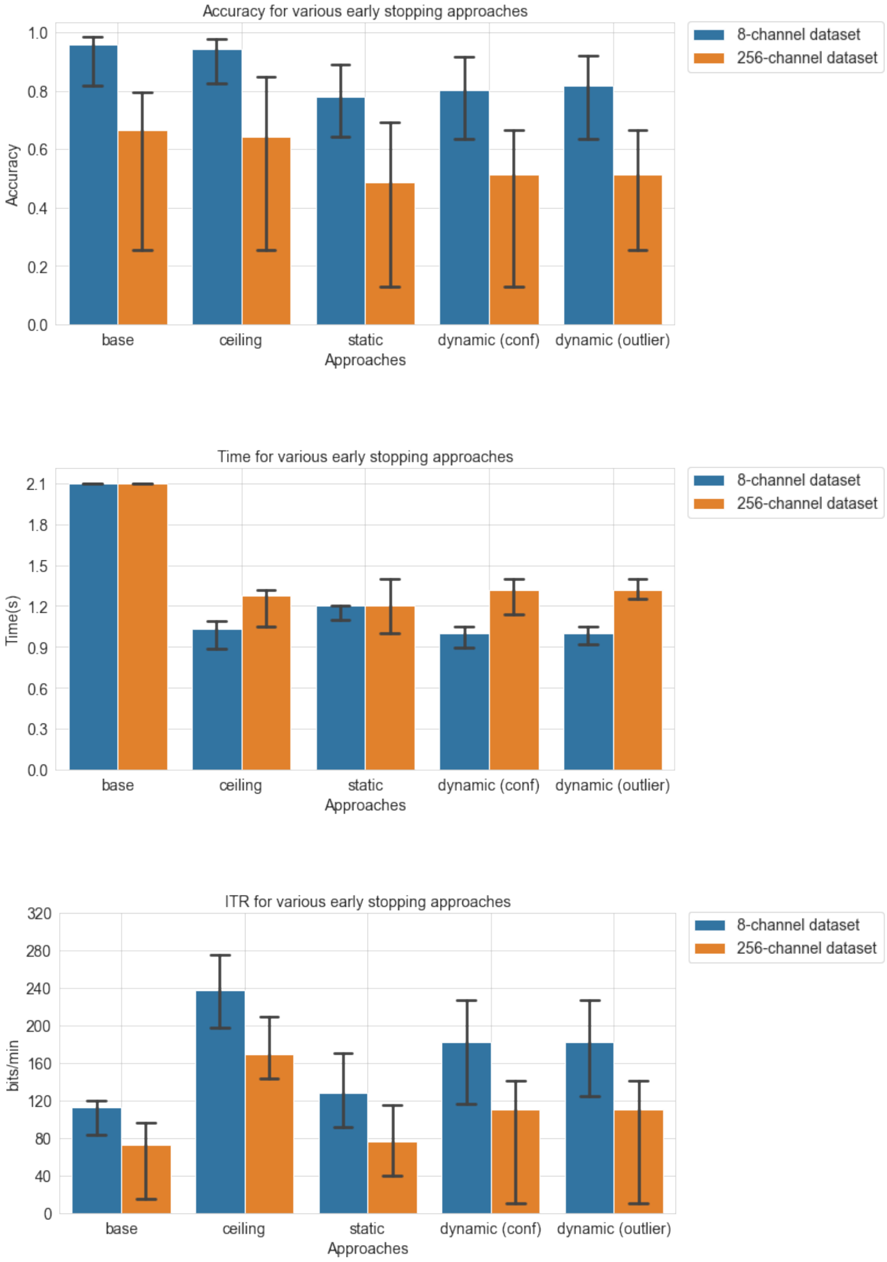


Figure 6.13: Performance metrics for early stopping

6.2 Comparison with other models

The following subsections detail the comparison in performance of the dual-objective CNN model to the other models namely CCA, EEG2Code and EEG-Inception over various evaluation metrics (accuracy, f1-score, information transfer rate, etc) during testing on both the 8-channel dataset and 256-channel dataset.

6.2.1 Accuracy

On the 8-channel dataset, the dual-objective CNN outperforms both CCA and EEG2Code in terms of within-subject and LOSO category accuracy. The improvement in accuracy is statistically significant on using a t-test whose p-values along with the accuracy differences as compared to the dual-objective CNN are reported in Table 6.6 with p-values that are significant being in bold. The dual-objective CNN model also outperforms EEG-Inception in terms of within-subject category accuracy, but the improvement in category accuracy obtained for the LOSO case was observed to be not significant after performing a t-test.

For the 256-channel dataset, the dual-objective CNN outperforms EEG-Inception in terms of both within-subject and LOSO category accuracy. The improvement in accuracy is statistically significant on using a t-test whose p-values along with the accuracy differences as compared to the dual-objective CNN are reported in Table 6.7 with p-values that are significant being in bold. However, the dual-objective CNN performs worse than CCA for the within-subject case and only outperforms EEG2Code for the LOSO case in terms of category accuracy. The CCA model obtained higher accuracies on the 256-channel dataset as compared to the dual-objective CNN, but the difference in accuracy was not significant for the LOSO case. The EEG2Code model obtained higher category accuracy as compared to the dual-objective CNN for the within-subject case but this difference proved to be not significant after performing a t-test. Section A.2.1 details the within-subject and LOSO category accuracy across models for individual subjects in both the datasets.

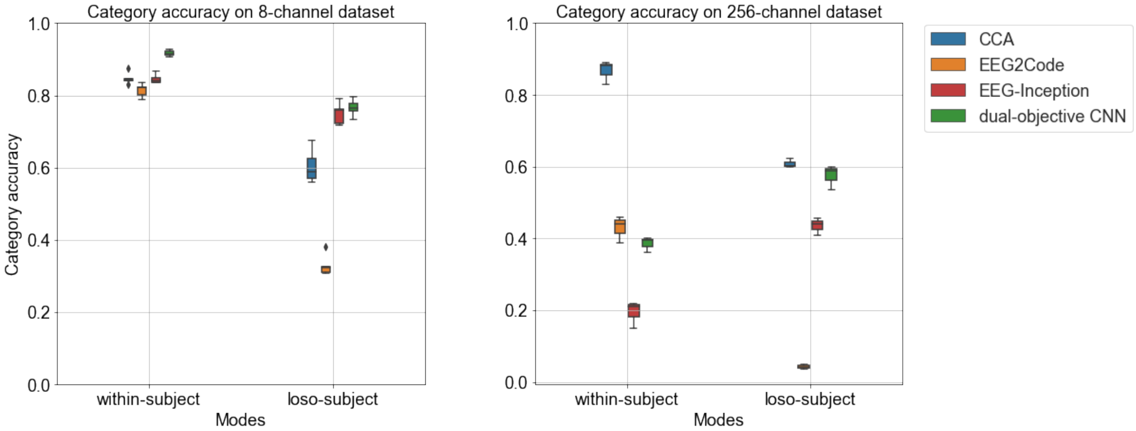


Figure 6.14: Mean category accuracy of all models across subjects

Models	Modes	accuracy comparison (%)	t-statistic	α	p-value
CCA	within-subject	-7.03	8.25	0.05	**0.0**
	LOSO	-16.19	6.84	0.05	**0.0**
EEG2Code	within-subject	-10.27	10.87	0.05	**0.0**
	LOSO	-43.66	25.64	0.05	**0.0**
EEG-Inception	within-subject	-7.14	9.84	0.05	**0.0**
	LOSO	-1.62	0.925	0.05	0.382

Table 6.6: Significance testing on category accuracy for 8-channel dataset between other models and dual-objective CNN using t-test

Models	Modes	accuracy comparison (%)	t-statistic	α	p-value
CCA	within-subject	+48.2	-21.73	0.05	**0.0**
	LOSO	+3.33	-1.59	0.05	0.186
EEG2Code	within-subject	+4.44	-1.84	0.05	0.14
	LOSO	-53.24	26.467	0.05	**0.0**
EEG-Inception	within-subject	-19.17	7.66	0.05	**0.0**
	LOSO	-13.98	5.78	0.05	**0.004**

Table 6.7: Significance testing on category accuracy for 256-channel dataset between other models and dual-objective CNN using t-test

6.2.2 F1-score and ITR

Since the datasets are balanced with the same distribution of corresponding target classes, the comparison of f1-scores obtained across models does not provide any additional information on accuracy as seen in Figure 6.15. The ITR across models also reflects the same information as accuracy. This is because the ITR when performing predictions accounts for the time corresponding to the duration of input data which is the same across all the models as the dual-objective CNN without dynamic stopping is compared with the other models. Section A.2.2 details the within-subject and LOSO ITR across models for individual subjects in both the datasets.

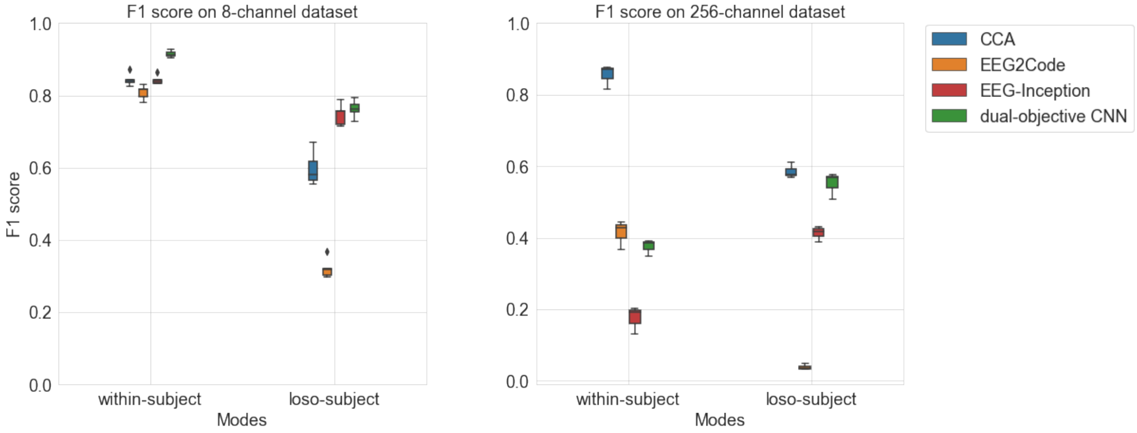


Figure 6.15: Mean f1-score of all models across subjects

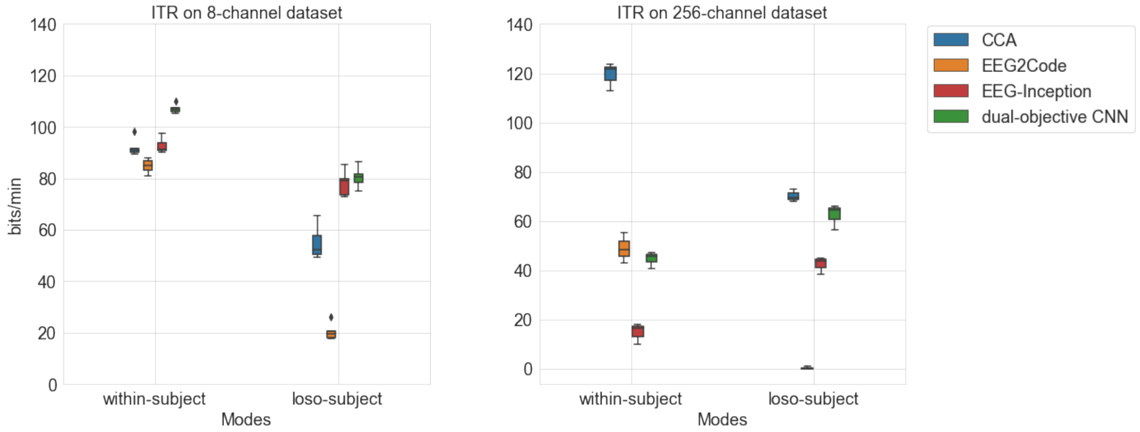


Figure 6.16: Mean ITR of all models across subjects

6.2.3 Performance over time-steps

Figure 6.18 depicts the comparison of accuracy and information transfer rate (ITR) between models over time. Section A.2.3 details the within-subject and LOSO category accuracy as well as ITR for individual subjects in both the datasets obtained by each of the models. The accuracy differences between the models that proved to be significant using a t-test are also reflected in the category accuracy at intermediate time points.

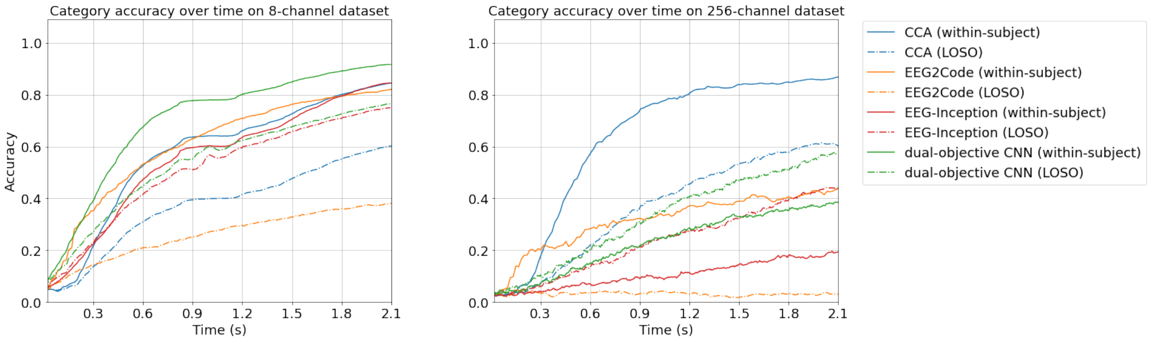


Figure 6.17: Mean category accuracy of all models over time across subjects

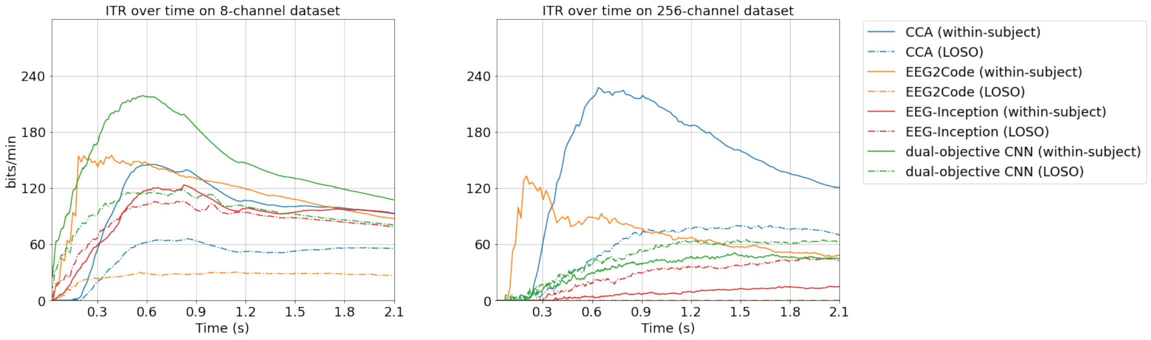


Figure 6.18: Mean ITR of all models over time across subjects

6.3 Discussion

This section summarizes the insights from the results obtained by the dual-objective CNN model in terms of the research questions posed in Subsection 1.1.

From Section 6.2, statistical significance testing using a t-test as seen in Tables 6.6 and 6.7 shows that the dual-objective CNN model outperforms CCA and EEG2Code for the within-subject and LOSO case whereas it outperforms EEG-Inception for the within-subject case on the 8-channel dataset as seen in Figure 6.14. However on the 256-channel dataset, the dual-objective CNN does not outperform the CCA model and outperforms the EEG2Code model only on the LOSO case as seen in Tables 6.6 and 6.7. Subsection 6.1.1 suggests that overfitting on the training data is the reason behind the poor generalization performance for the within-subject case on the 256-channel dataset as depicted in 6.2. The reasons behind this overfitting were further elaborated in Subsection 6.1.2. Nevertheless, on the 8-channel dataset, the dual-objective CNN model still significantly outperforms flash-level predictive models as the neural network model is designed to perform predictions at the level of trials (.i.e. stimulation sequence level). The performance improvement is in terms of both accuracy and speed (.i.e by incorporating dynamic stopping), thereby obtaining a better ITR as well.

The normalized confusion matrices depicted in Subsections 6.1.5 and A.1.4 with high values along the diagonal indicate that most of the predictions made my the model are either true positive or true negative. Further examination of the off-diagonal values of the confusion matrices shows no outlier patterns in terms of false positives and false negatives among the target classes of stimulation sequences.

Subsection 5.3.2 details the control-state detection for the 8-channel dataset which allows the model to work as an asynchronous system. The non-control state data being added as an extra target class (.i.e class 21 for the 8-channel dataset) permits the system to start a new trial without emitting a prediction relating to a symbol on the screen when the subject is not overtly attending to the task. The normalized confusion matrices depicted in Subsections 6.1.5 and A.1.4 shows a high value (i.e close to 1) for the non-control state data. This indicates that the model is able to differentiate between the non-control state data and the control-state data with high accuracy. The ROC-curves illustrated in Subsection A.1.5 shows a high AUC score (area under the curve) for the target class corresponding to the non-control state data in the 8-channel dataset. This indicates that the proposed model allows for high separability between the control-state data and the non-control state data.

The explainability of the model in terms of spatial and temporal patterns are depicted in Subsection 6.1.6. The spatial patterns are representative of the channels in the recorded data that contributed most to the extracted spatial features. As expected, the spatial patterns are concentrated near the occipital region of the head where the primary visual cortex lies that processes visual information which is relayed by the retinas as seen in Subsection 6.1.6.1. The Grad-CAM algorithm was used to obtain heatmaps that provided insights into the regions of temporal importance within the data thereby revealing temporal patterns that were most informative for the model in classifying the data. From the visualized temporal patterns as seen in Subsection 6.1.6.2, it is evident that the model gives importance to the first and last regions of the 2.1 window, whereas it does not learn any temporal information from the region in between 0.8 and 1.1s for the 8-channel dataset. This gap in temporal information was further elaborated in Subsection 6.1.6.1. For the 256-channel dataset, regions of temporal importance were observed intermittently along the 2.1s trial and no such prolonged gap in temporal information was observed.

The extent of fine-tuning required to adapt the LOSO dual-objective CNN model trained on a population of subjects to a specific unseen subject is depicted in Subsection 6.1.7. The proposed model was adapted to data from an unseen subject by first freezing all the layers in the model except the output layers (.i.e softmax and sigmoid layers) which are re-trained on validation data of the corresponding subject. For both the datasets, the proposed model required around 20 additional trials (each of duration 2.1s) to adapt its weight parameters for optimizing subject-specific performance.

Dynamic stopping allowed the dual-objective CNN model to emit target class predictions at much shorter durations than the input data duration of 2.1s as seen in Subsection 6.1.8. From the performance metrics (accuracy, time and ITR) for various early stopping approaches, it was observed that the performance of the proposed model improved on the base performance (i.e. without dynamic stopping). On comparing the performance between the static stopping and dynamic stopping approaches, it was evident that the dynamic stopping approaches obtained higher accuracies compared to static stopping. Since the time required for obtaining those accuracies were lower for the 8-channel dataset and comparable for the 256-channel dataset, the ITR for the dynamic stopping approaches were higher as compared to static stopping. Both the dynamic stopping methods based on confidence thresholding and outlier detection respectively showed similar performance.

Chapter 7

Conclusions and Future Work

Deep learning has proved its potential for decoding full stimulation sequences directly from c-VEP response data through this research. Convolution neural networks trained on a dual-objective function for both decoding the bits in the stimulation sequence and classifying the target class of stimulation sequence simultaneously significantly improved on the performance of correlation based techniques that decoded at the flash-level and neural networks with a single objective of classifying the target classes using only a softmax function in the output layer. The performance improvement is not only limited in terms of accuracy but also the required time (i.e. by incorporating dynamic stopping) and thereby the information transfer rate(ITR) as well. The speed of classification is improved in the proposed model by using a masking layer in the first layer which along with the dynamic stopping rule enables the model to emit a prediction in durations shorter than the trial time of 2.1s. The model can also be adapted for subject-specific performance by using transfer learning. The spatial and temporal patterns obtained using the learned weight parameters also gives insights into the corresponding patterns in c-VEP response data.

The disadvantages of the model include sensitivity to noisy data where the model suffers loss in performance due to overfitting on the noise present in the training data especially for the 256-channel dataset. This could be improved by extending the data prepossessing pipeline of the model. Although separating the spatial and temporal feature extraction layers in the network allows for visualizing intuitive spatial and temporal patterns in the data, it could create a bottleneck in the network. The gap in temporal information (0.8s-1.1s) for the 8-channel dataset was consistent among all the models (CCA, EEG2Code, EEG-Inception and dual-objective CNN) indicating that this gap in temporal information is inherent to the dataset and not model-specific. However, a conclusive answer to the reason behind this gap in temporal information in the 8-channel dataset could not be attained.

The proposed model could also be further improved by modifying the model architecture using Recurrent neural networks or Long short term memory units (LSTMs) which uses a vector of hidden variables as memory to capture information from the past for making current and future predictions. However, the naive implementation of LSTMs in the model architecture for temporal feature extraction did not significantly improve the performance of the proposed model and hence was not included in the obtained results. Generative adversarial networks (GANs) could also improve the sensitivity of the network to noisy data and obtain higher performance for the subjects that the proposed model was unable to generalize to. The data preprocessing part of the pipeline could also be improved by incorporating it into an end-to-end model where the learned parameters are used in the data preprocessing stage.

Appendix A

Appendix

A.1 Dual-objective CNN (Results)

A.1.1 Accuracy

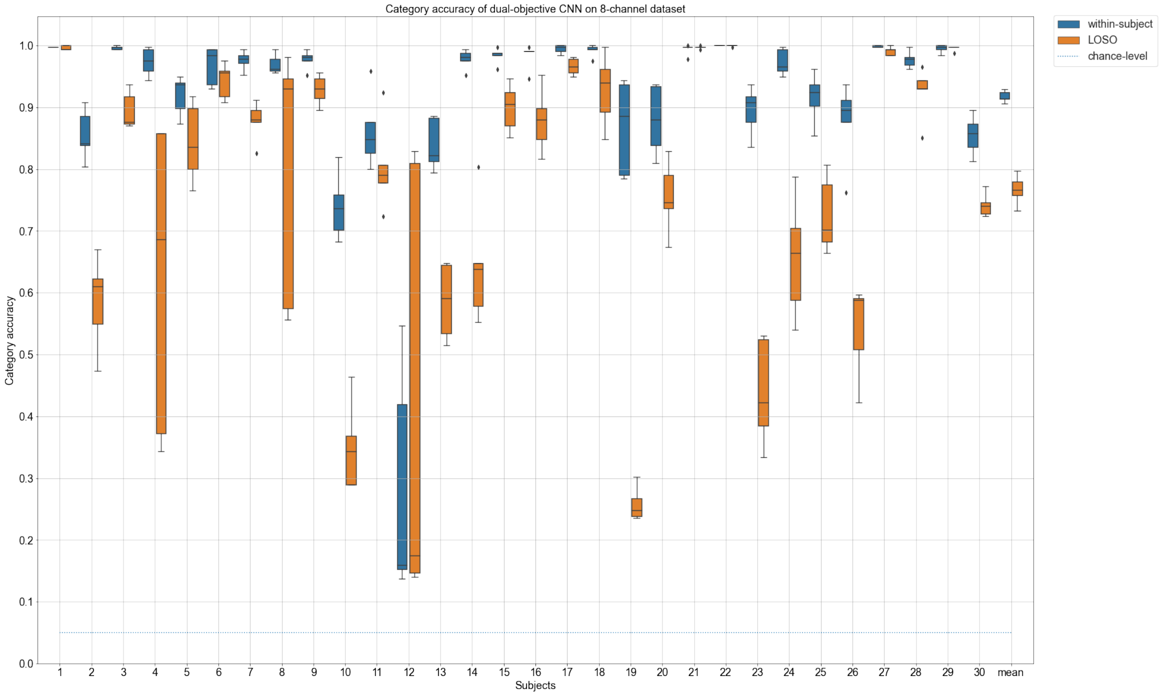


Figure A.1: Category accuracy of dual-objective CNN on 8-channel dataset

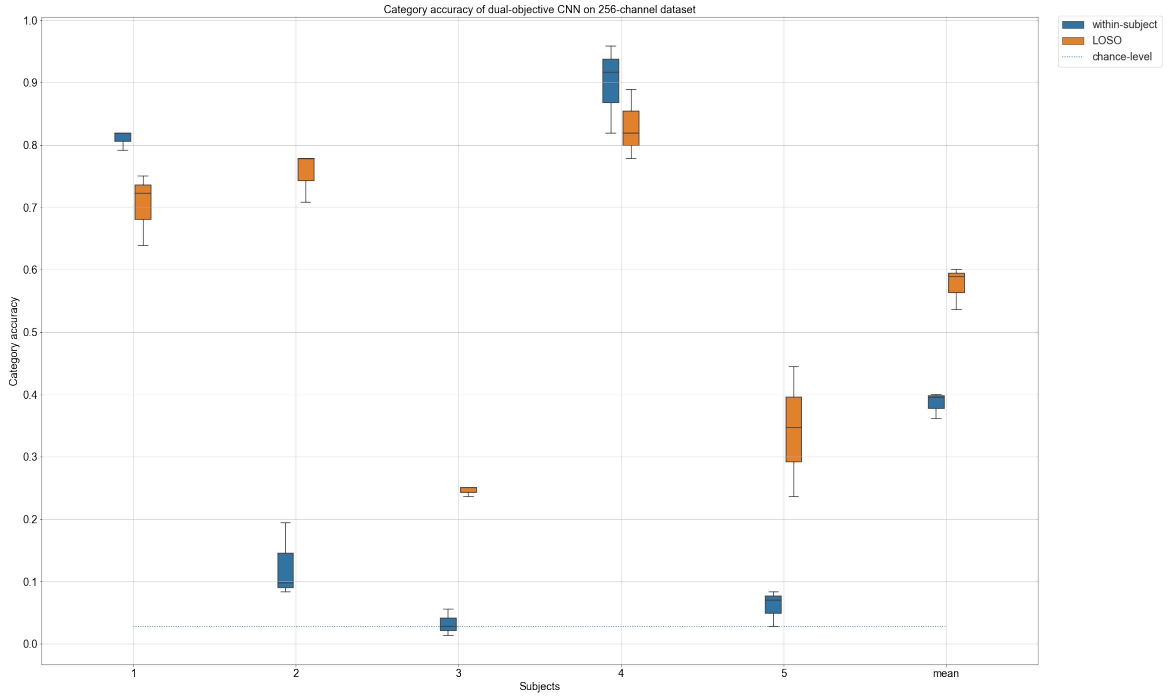


Figure A.2: Category accuracy of dual-objective CNN on 256-channel dataset

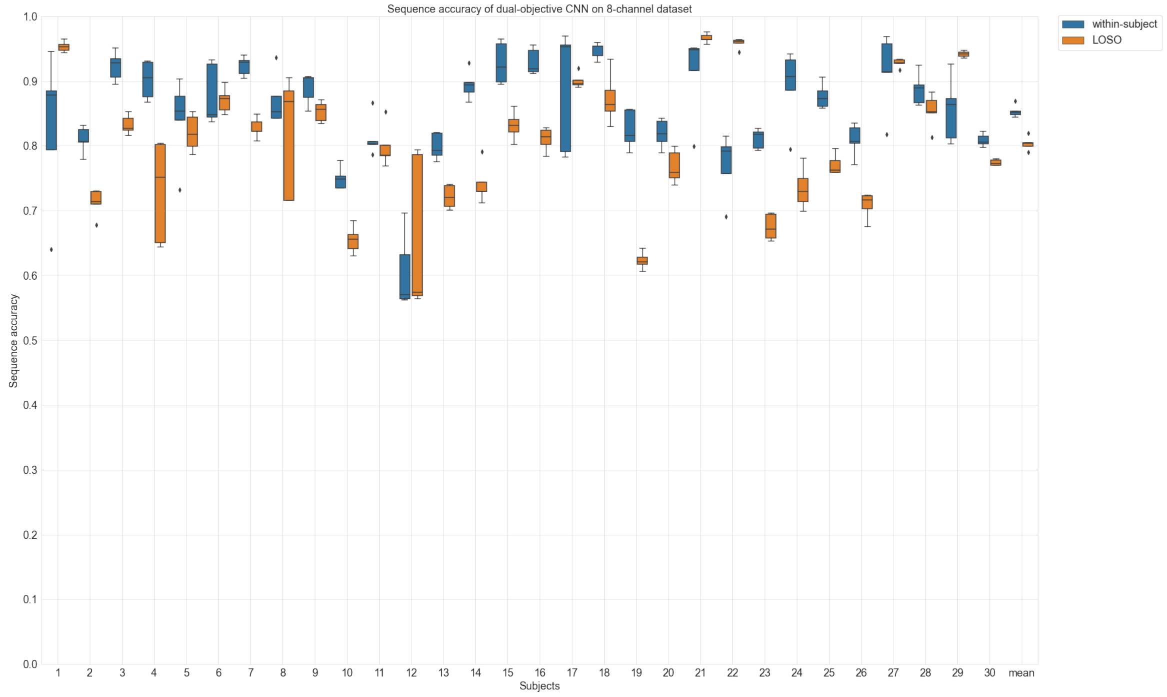


Figure A.3: Sequence accuracy of dual-objective CNN on 8-channel dataset

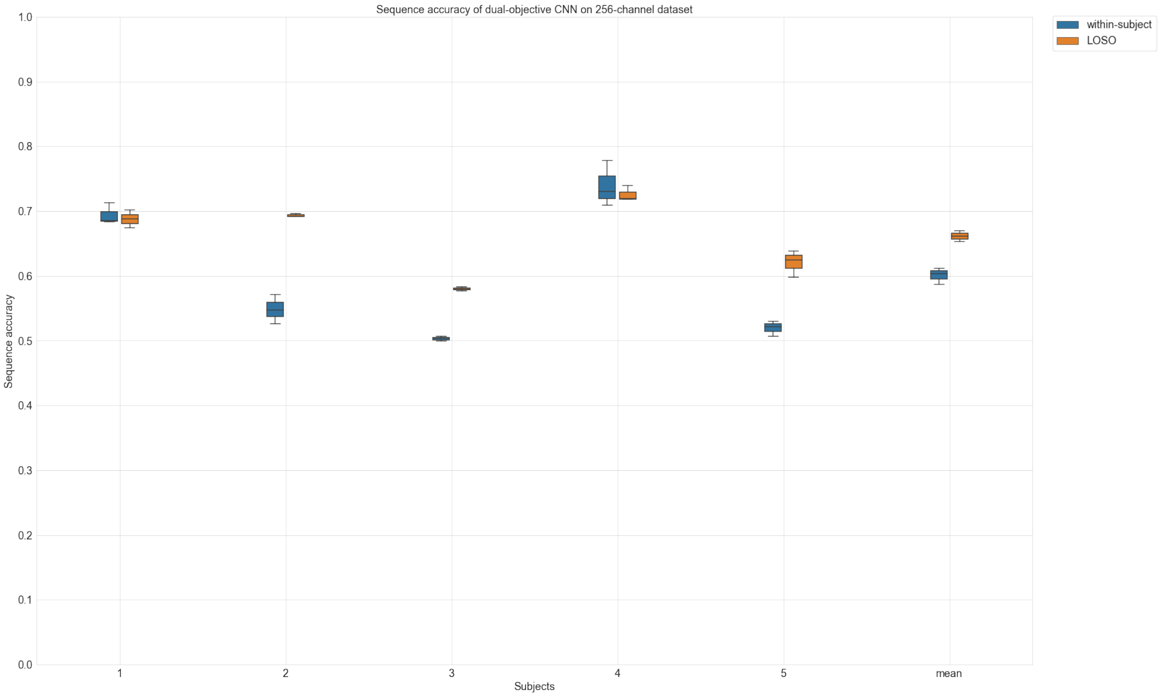


Figure A.4: Sequence accuracy of dual-objective CNN on 256-channel dataset

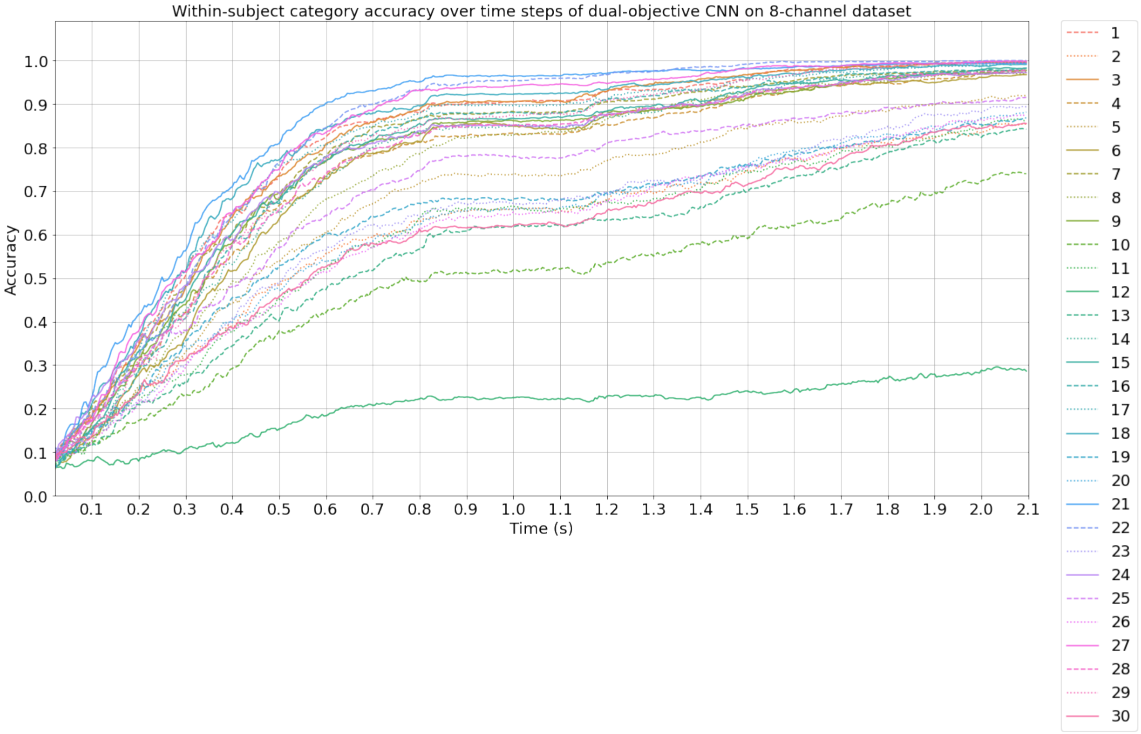


Figure A.5: Within-subject accuracy over time-steps of dual-objective CNN on 8-channel dataset

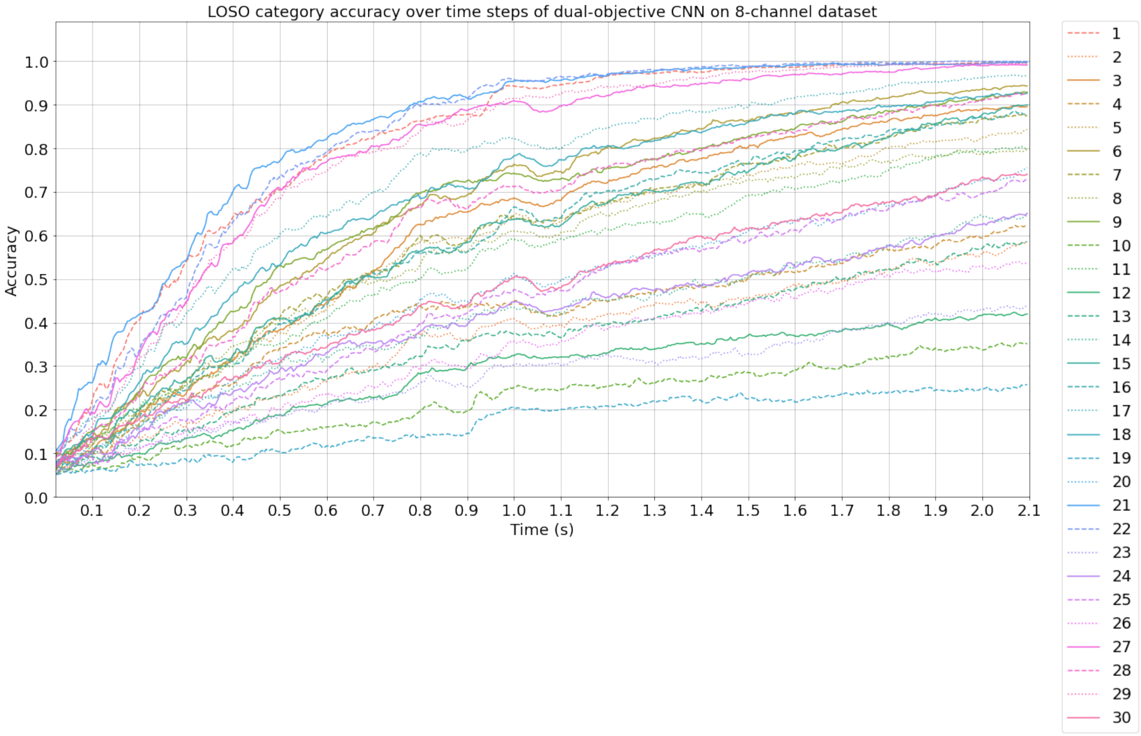


Figure A.6: LOSO accuracy over time-steps of dual-objective CNN on 8-channel dataset

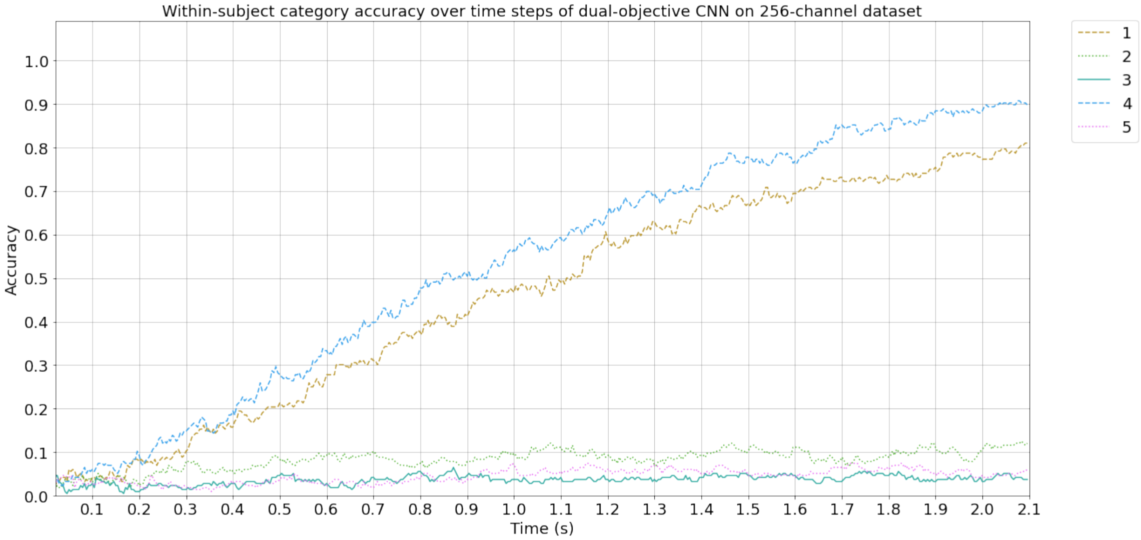


Figure A.7: Within-subject accuracy over time-steps of dual-objective CNN on 256-channel dataset

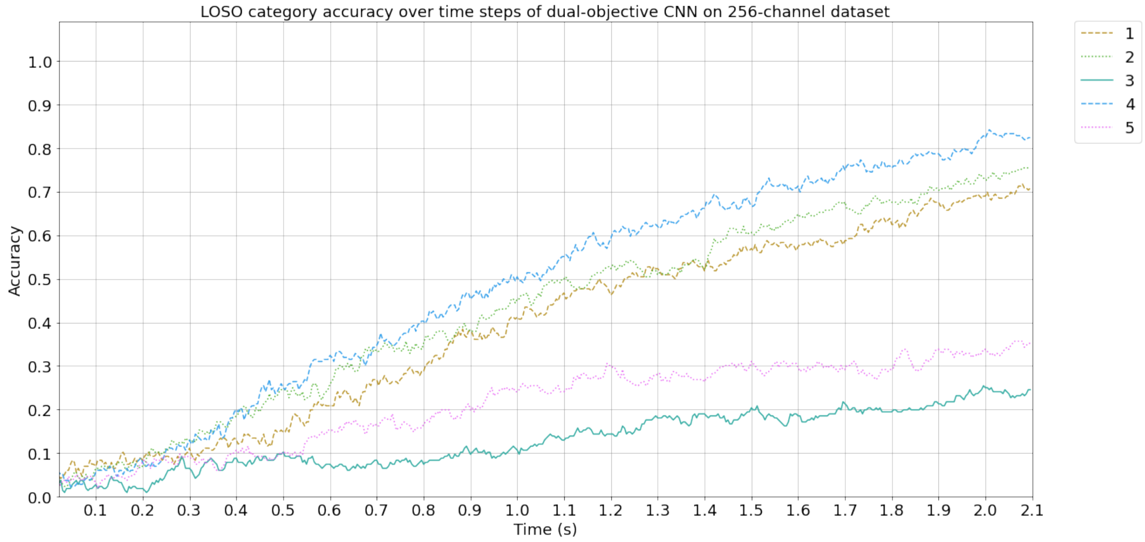


Figure A.8: LOSO accuracy over time-steps of dual-objective CNN on 256-channel dataset

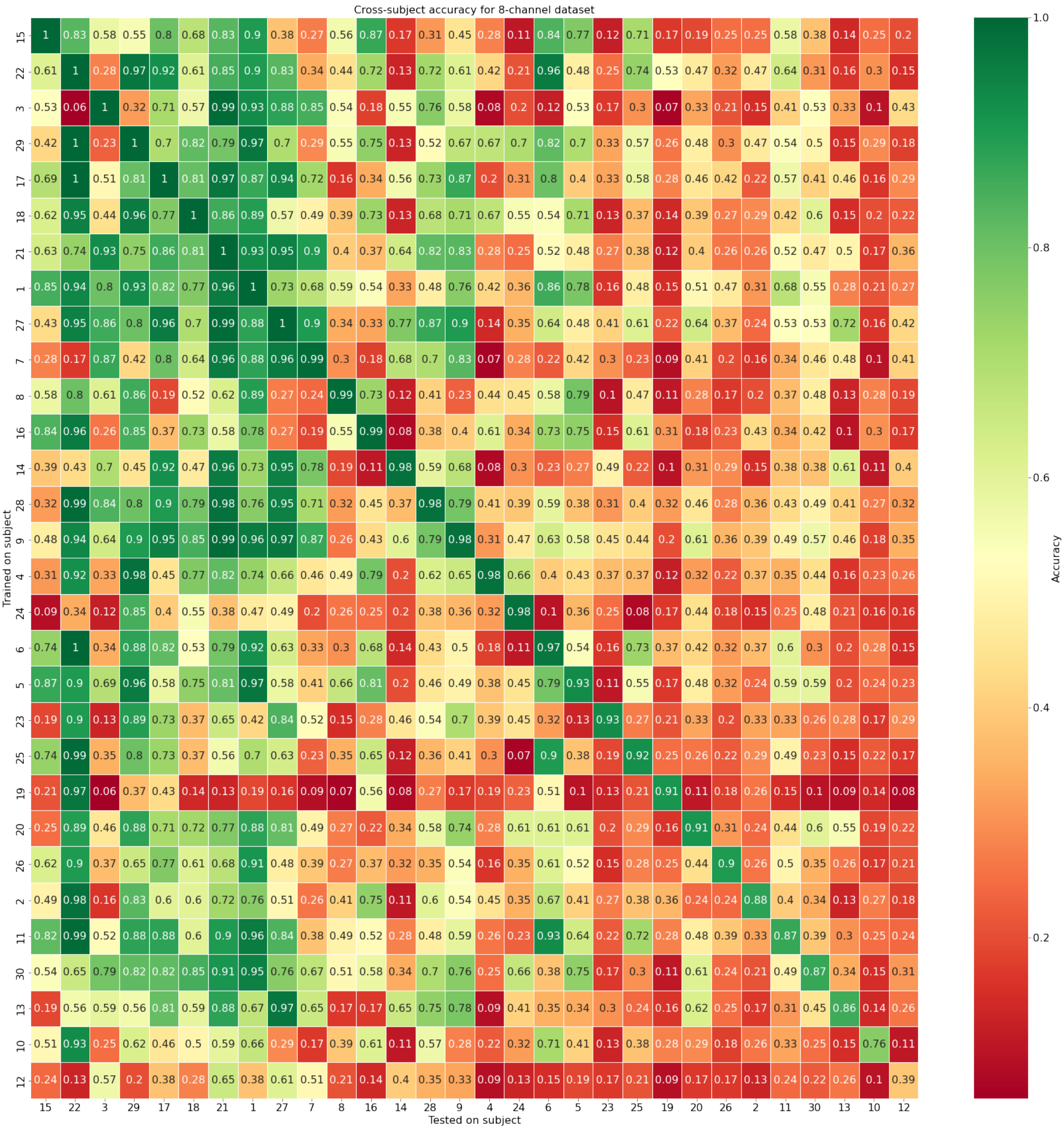


Figure A.9: Cross-subject accuracy of dual-objective CNN on 8-channel dataset

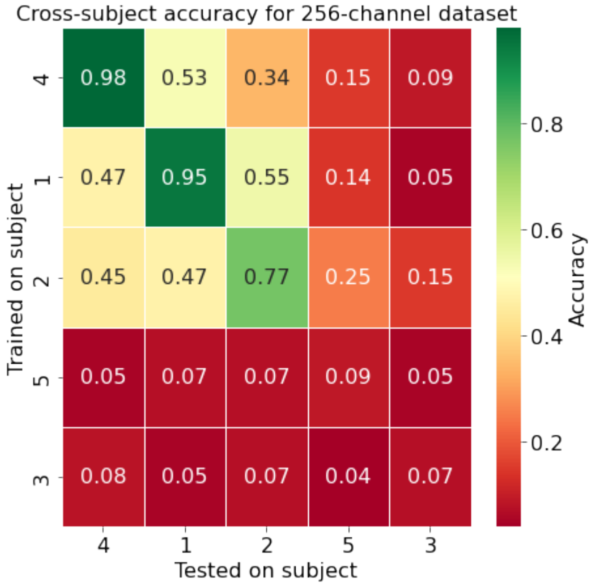


Figure A.10: Cross-subject accuracy of dual-objective CNN on 256-channel dataset

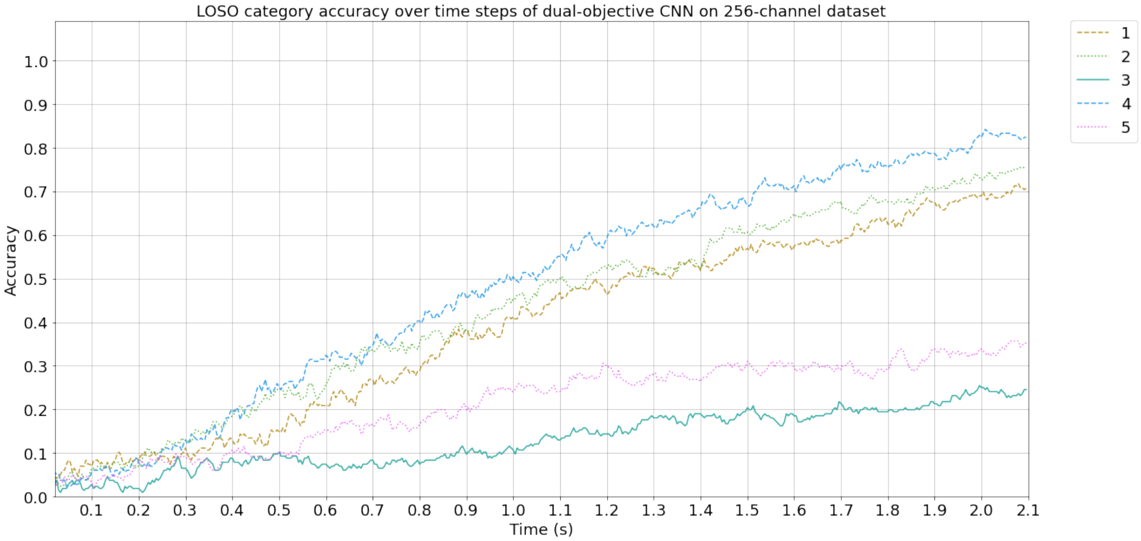


Figure A.11: LOSO accuracy over time-steps of dual-objective CNN on 256-channel dataset

A.1.2 Information transfer rate(ITR)

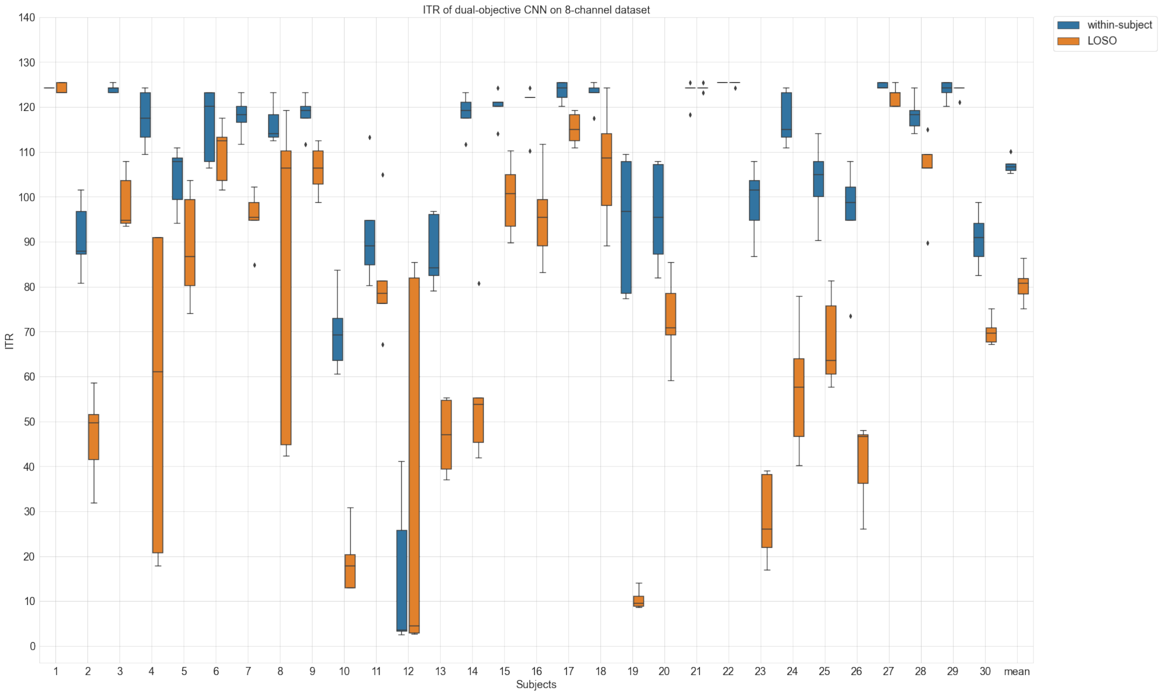


Figure A.12: ITR of dual-objective CNN on 8-channel dataset

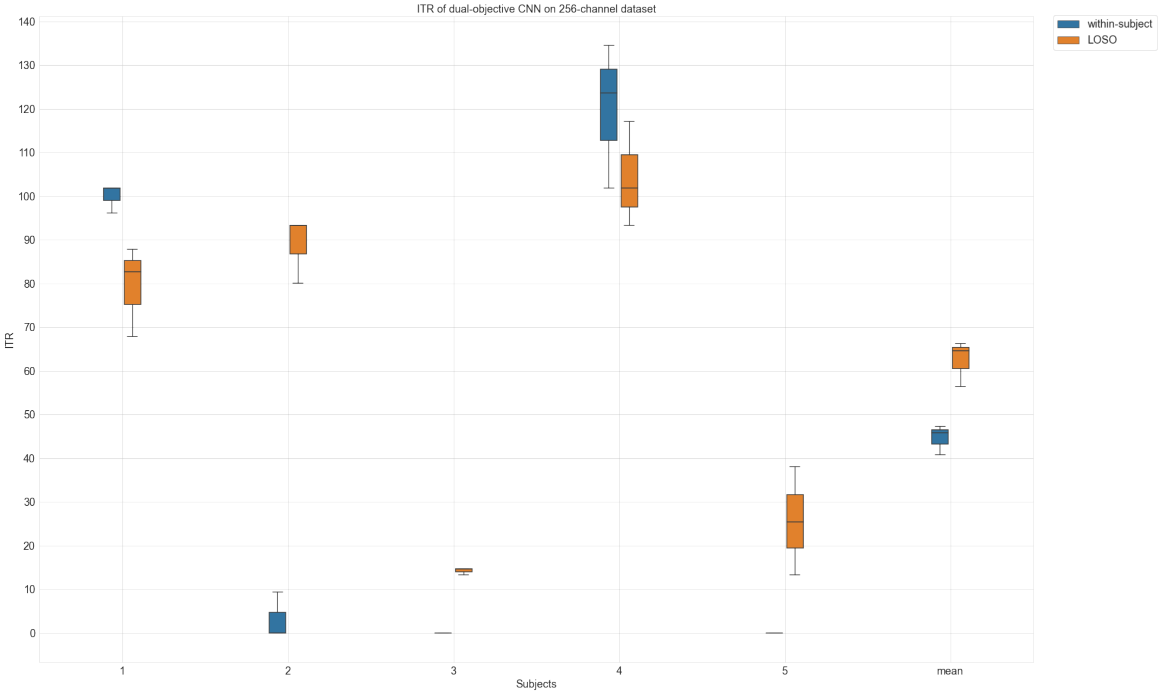


Figure A.13: ITR of dual-objective CNN on 256-channel dataset

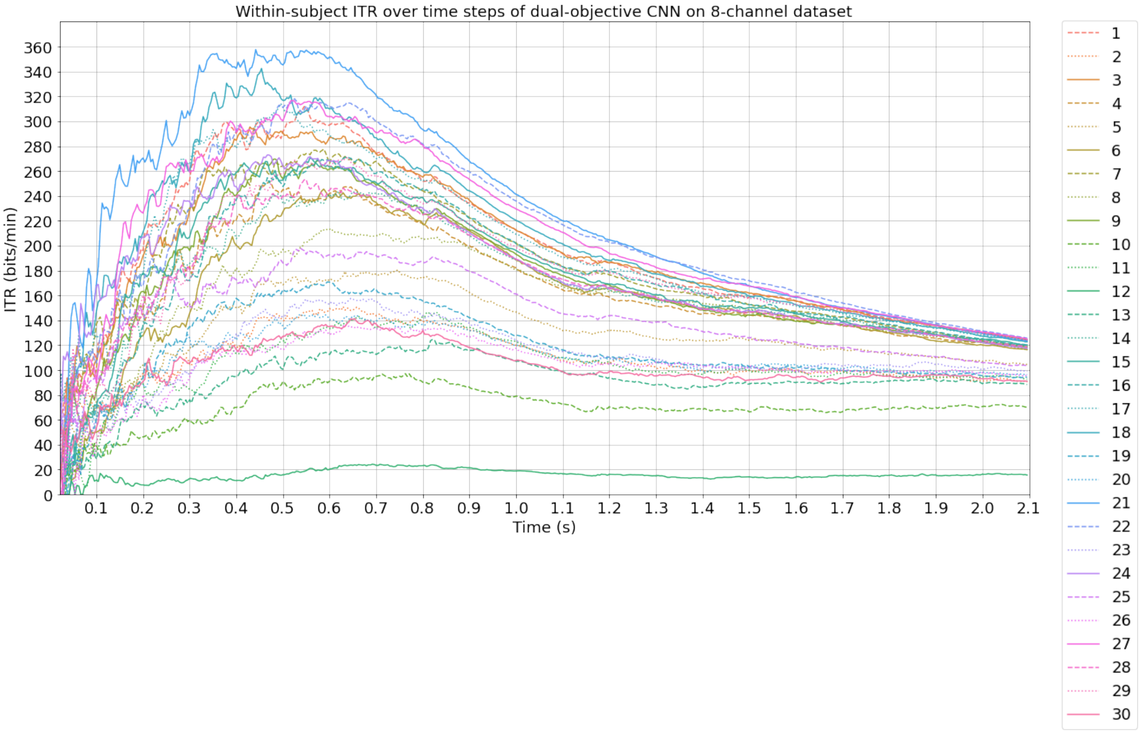


Figure A.14: Within-subject ITR over time-steps of dual-objective CNN on 8-channel dataset

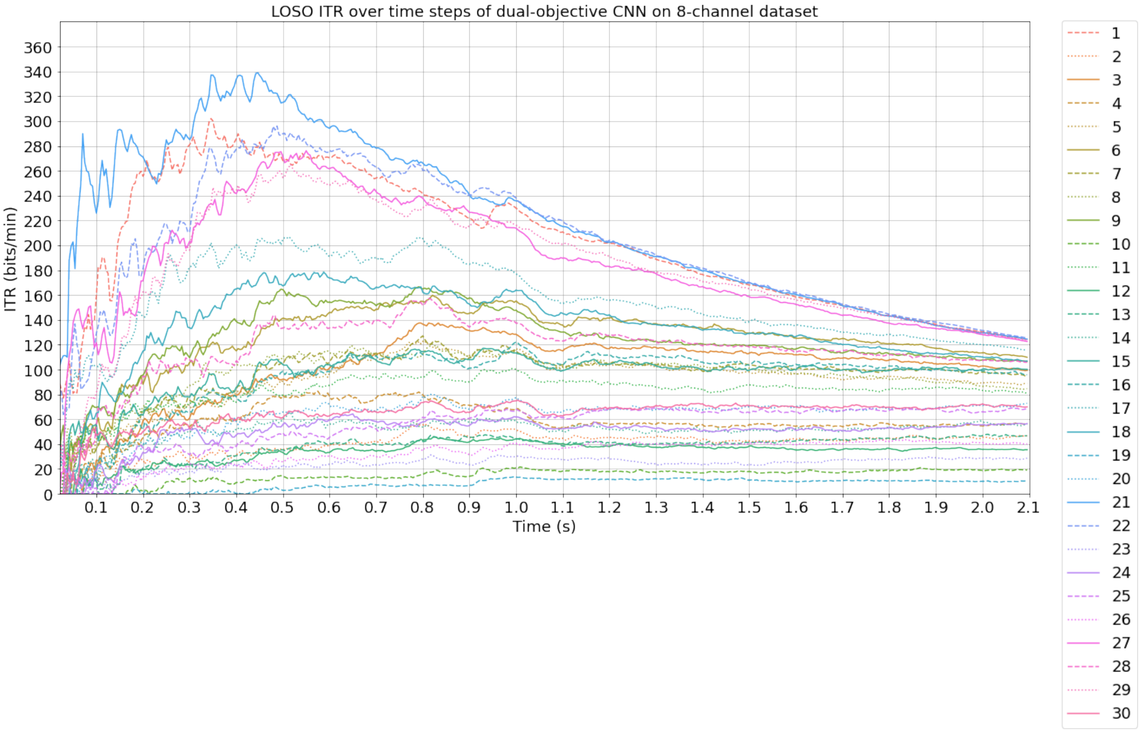


Figure A.15: LOSO ITR over time-steps of dual-objective CNN on 8-channel dataset

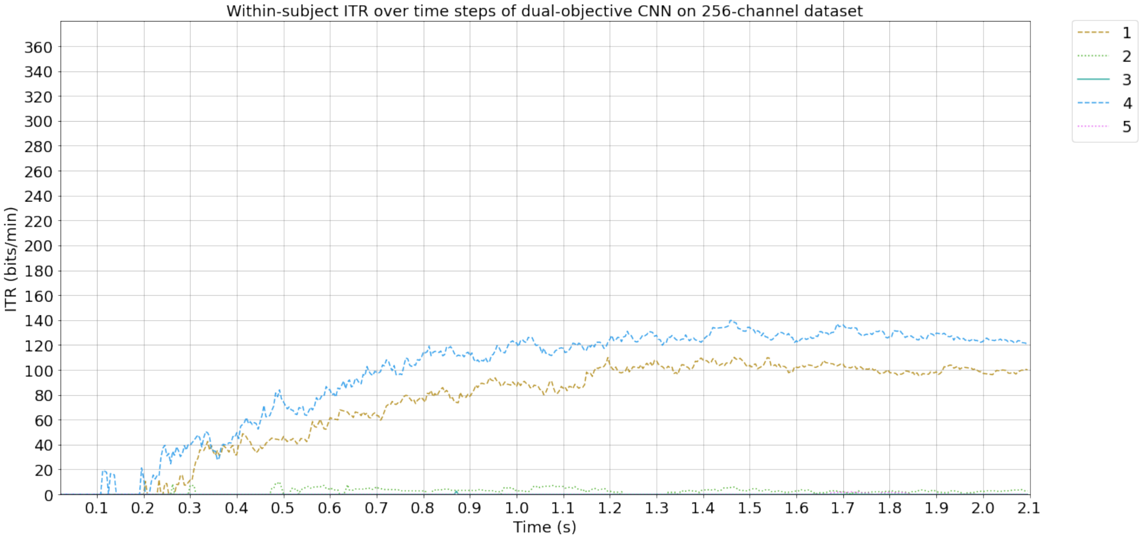


Figure A.16: Within-subject ITR over time-steps of dual-objective CNN on 256-channel dataset

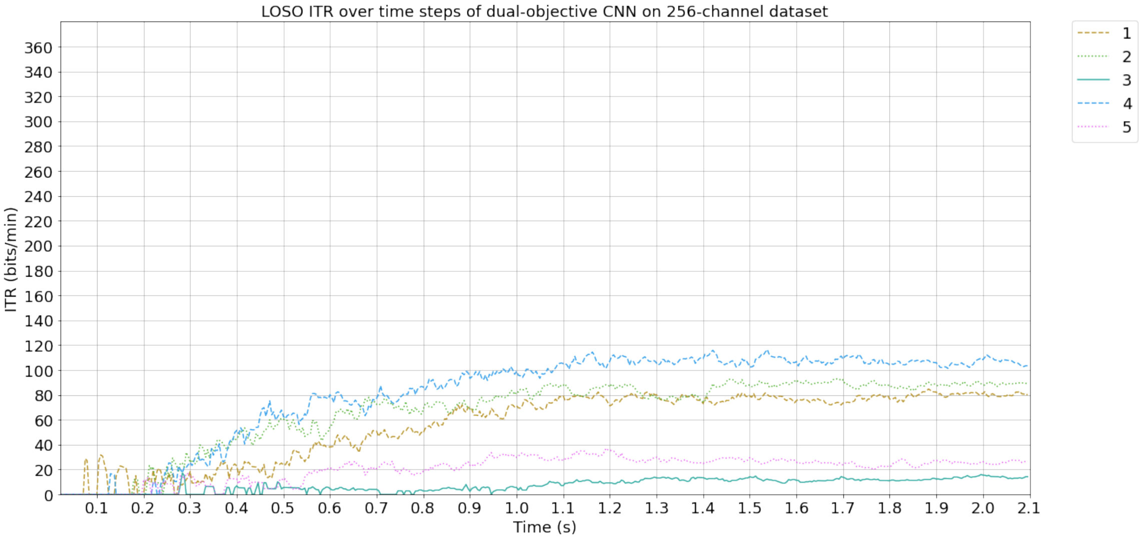


Figure A.17: LOSO ITR over time-steps of dual-objective CNN on 256-channel dataset

A.1.3 F1 score

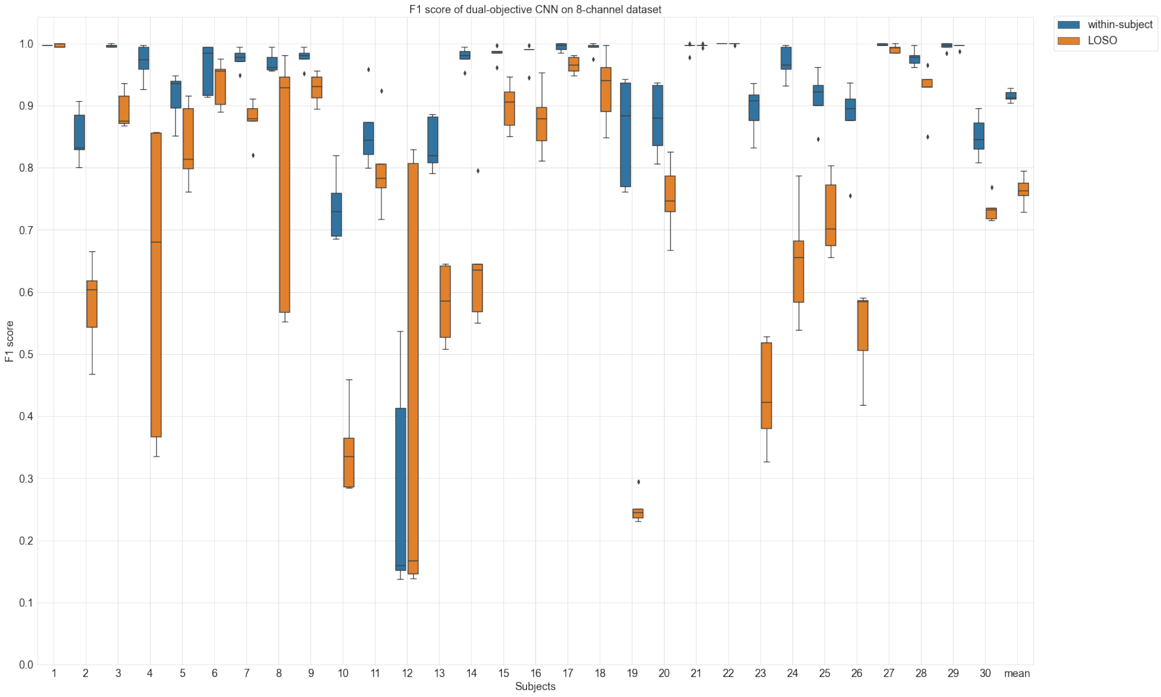


Figure A.18: F1 score of dual-objective CNN on 8-channel dataset

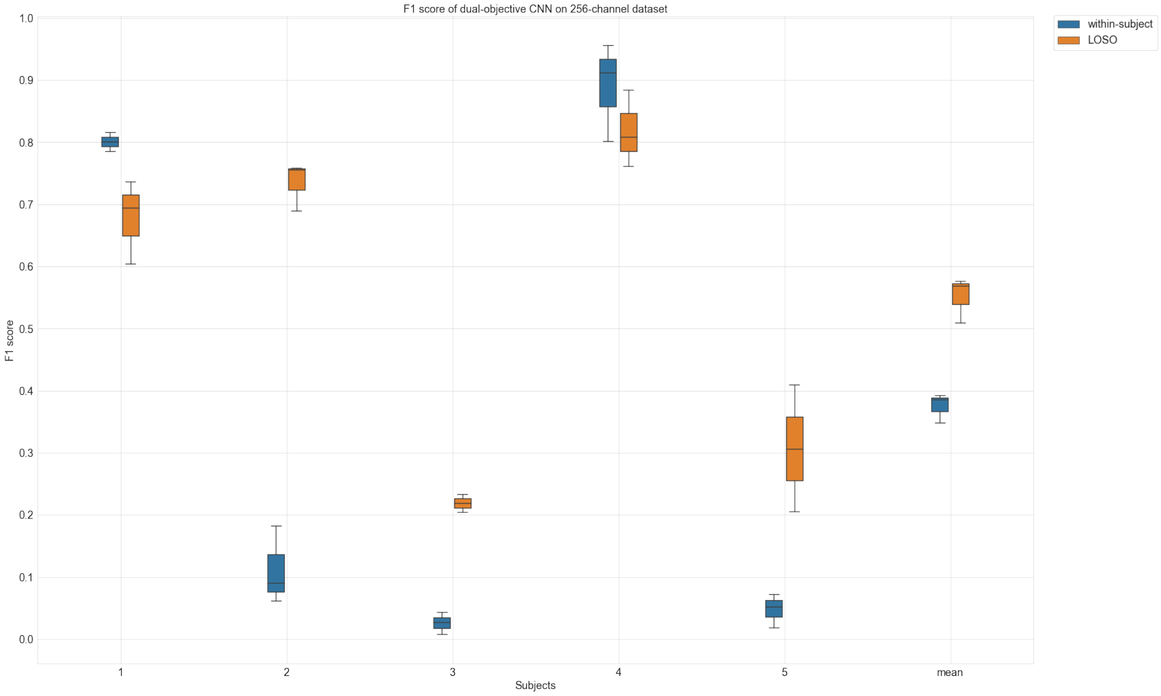
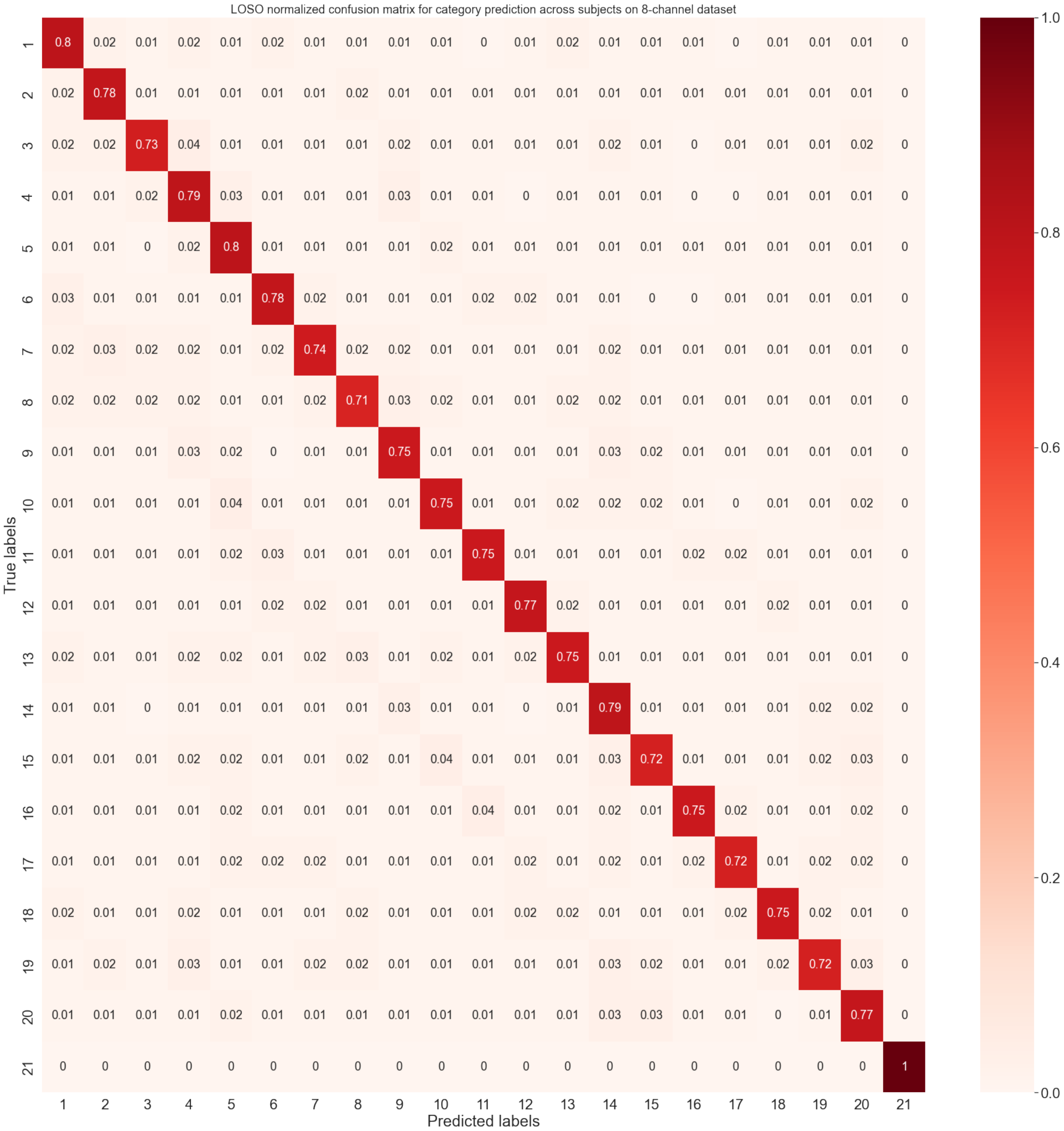


Figure A.19: F1 score of dual-objective CNN on 256-channel dataset

A.1.4 Confusion matrix



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	0.8	0.02	0.01	0.02	0.01	0.02	0.01	0.01	0.01	0.01	0	0.01	0.02	0.01	0.01	0.01	0	0.01	0.01	0.01	0
2	0.02	0.78	0.01	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0
3	0.02	0.02	0.73	0.04	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0	0.01	0.01	0.01	0.02	0
4	0.01	0.01	0.02	0.79	0.03	0.01	0.01	0.01	0.03	0.01	0.01	0	0.01	0.01	0.01	0	0	0.01	0.01	0.01	0
5	0.01	0.01	0	0.02	0.8	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0
6	0.03	0.01	0.01	0.01	0.01	0.78	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0	0	0.01	0.01	0.01	0.01	0
7	0.02	0.03	0.02	0.02	0.01	0.02	0.74	0.02	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0
8	0.02	0.02	0.02	0.02	0.01	0.01	0.02	0.71	0.03	0.02	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0
9	0.01	0.01	0.01	0.03	0.02	0	0.01	0.01	0.75	0.01	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.01	0.01	0.01	0
10	0.01	0.01	0.01	0.01	0.04	0.01	0.01	0.01	0.01	0.75	0.01	0.01	0.02	0.02	0.02	0.01	0	0.01	0.01	0.02	0
11	0.01	0.01	0.01	0.01	0.02	0.03	0.01	0.01	0.01	0.01	0.75	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0
12	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.77	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0
13	0.02	0.01	0.01	0.02	0.02	0.01	0.02	0.03	0.01	0.02	0.01	0.02	0.75	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0
14	0.01	0.01	0	0.01	0.01	0.01	0.01	0.01	0.03	0.01	0.01	0	0.01	0.79	0.01	0.01	0.01	0.01	0.02	0.02	0
15	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.02	0.01	0.04	0.01	0.01	0.01	0.03	0.72	0.01	0.01	0.01	0.02	0.03	0
16	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.04	0.01	0.01	0.02	0.01	0.75	0.02	0.01	0.01	0.02	0
17	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.01	0.01	0.01	0.01	0.02	0.01	0.02	0.01	0.02	0.72	0.01	0.02	0.02	0
18	0.02	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.02	0.75	0.02	0.01	0
19	0.01	0.02	0.01	0.03	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.01	0.01	0.03	0.02	0.01	0.01	0.02	0.72	0.03	0
20	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.03	0.03	0.01	0.01	0	0.01	0.77	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure A.20: LOSO normalized confusion matrix for category prediction on 8-channel dataset

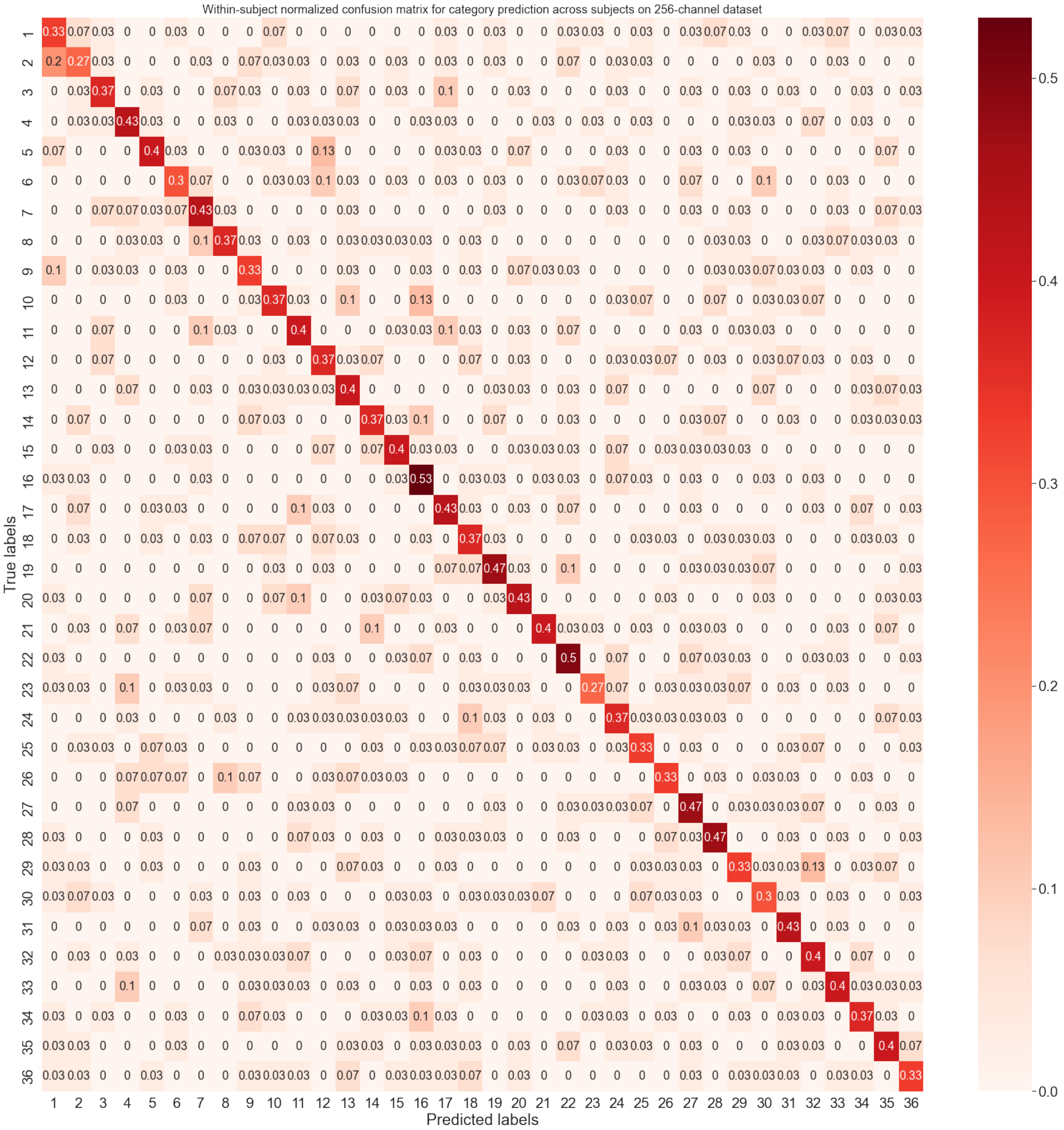


Figure A.21: Within-subject normalized confusion matrix for category prediction on 256-channel dataset

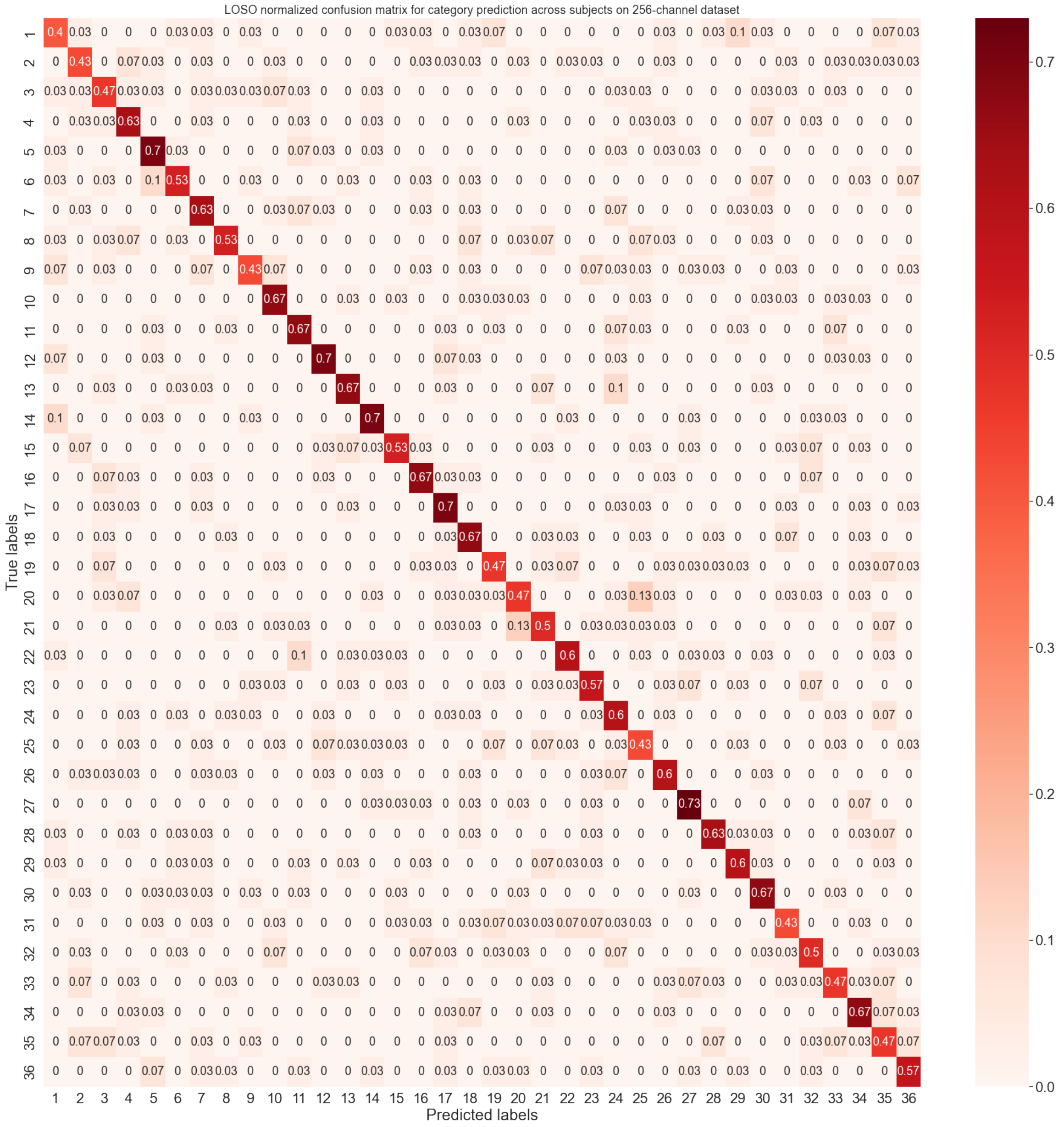


Figure A.22: LOSO normalized confusion matrix for category prediction on 256-channel dataset

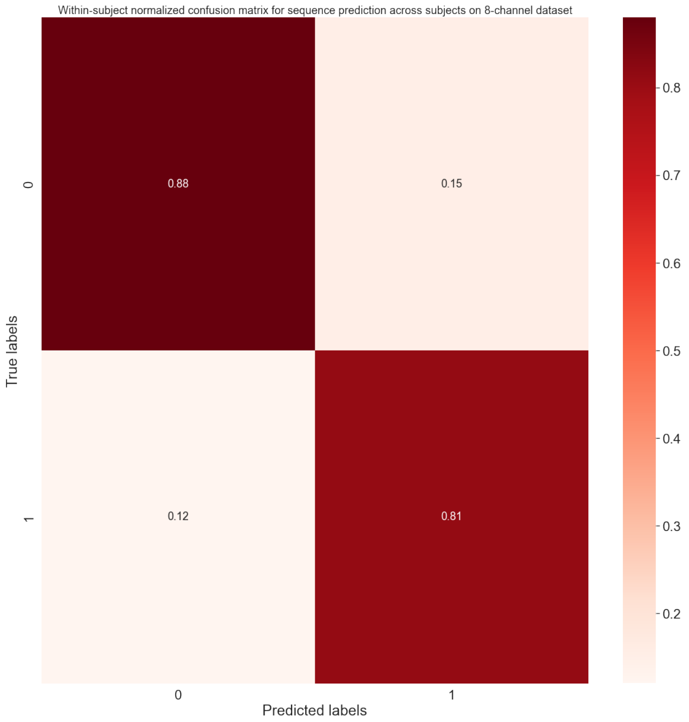


Figure A.23: Within-subject normalized confusion matrix for sequence prediction on 8-channel dataset

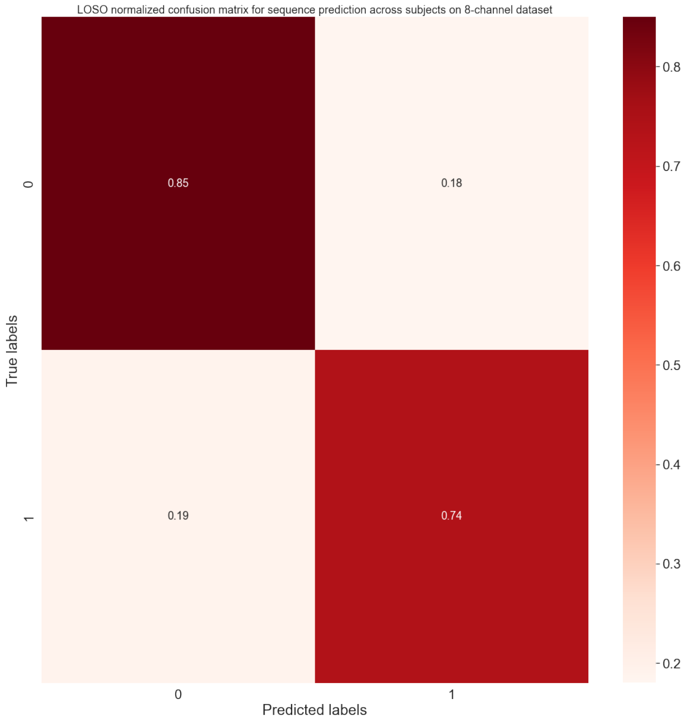


Figure A.24: LOSO normalized confusion matrix for sequence prediction on 8-channel dataset

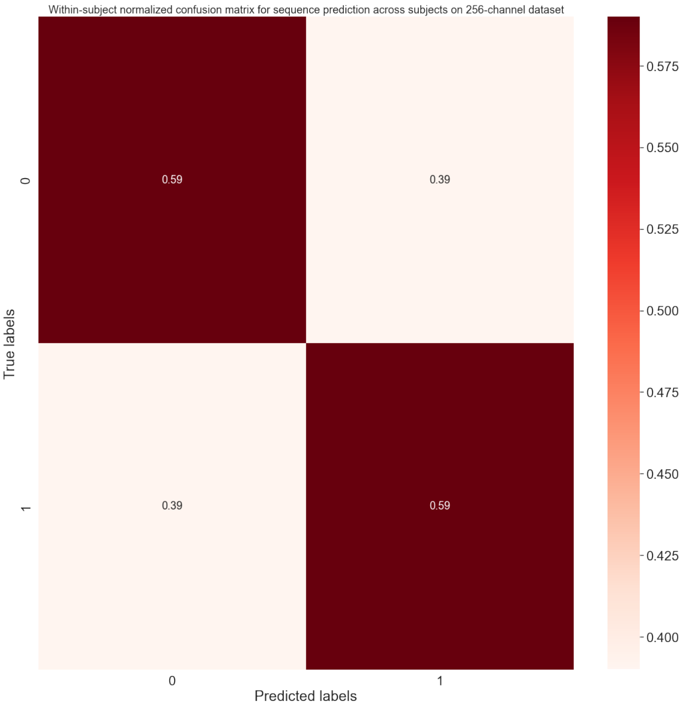


Figure A.25: Within-subject normalized confusion matrix for sequence prediction on 256-channel dataset

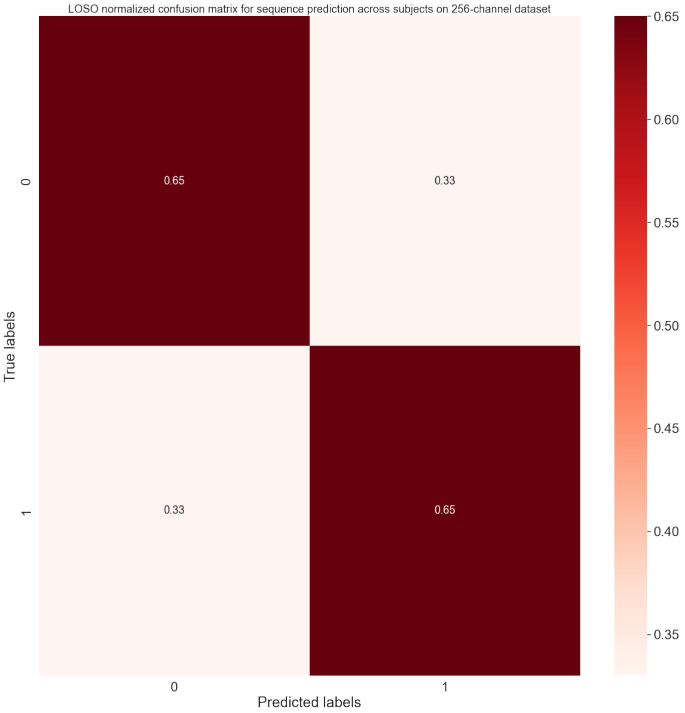
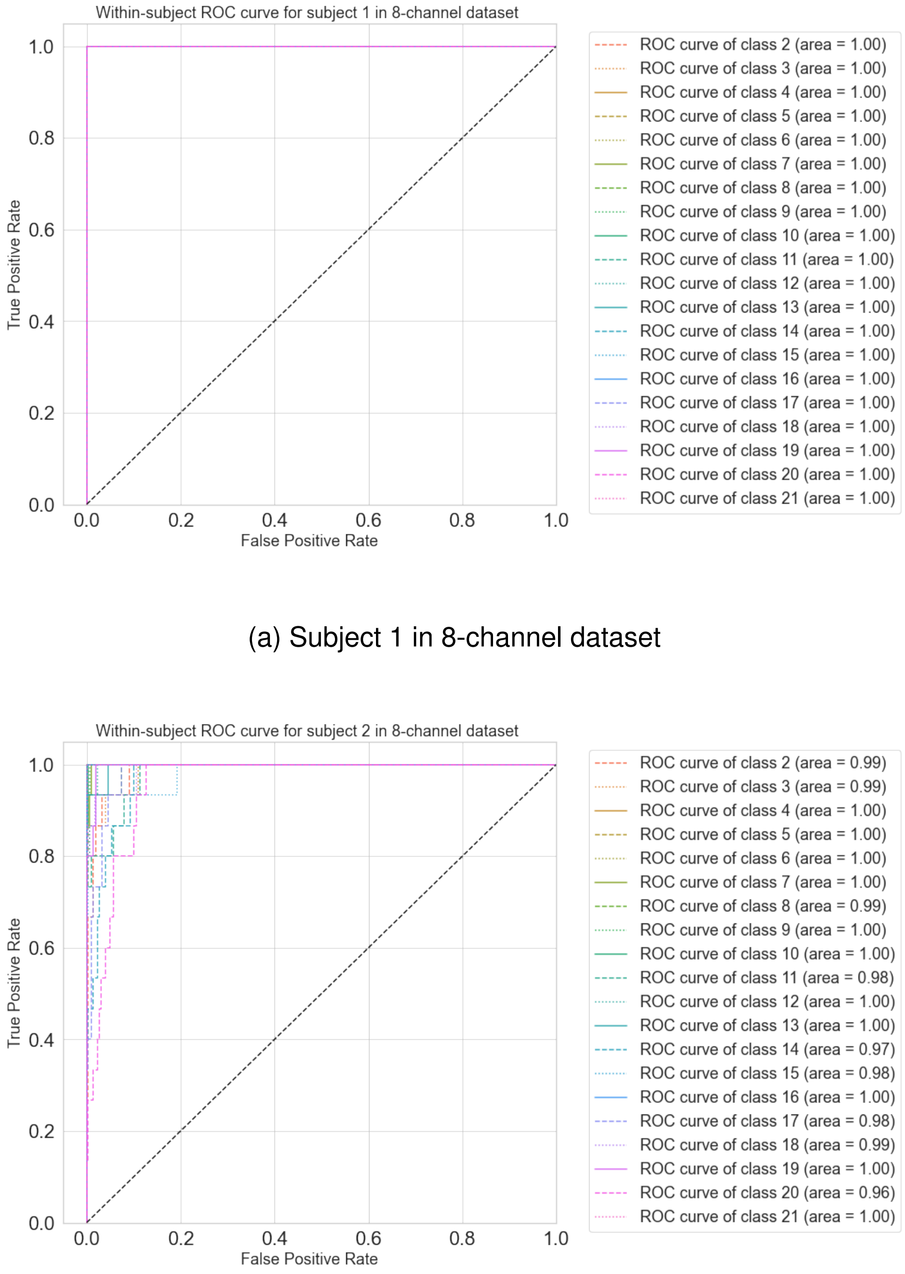


Figure A.26: LOSO normalized confusion matrix for sequence prediction on 256-channel dataset

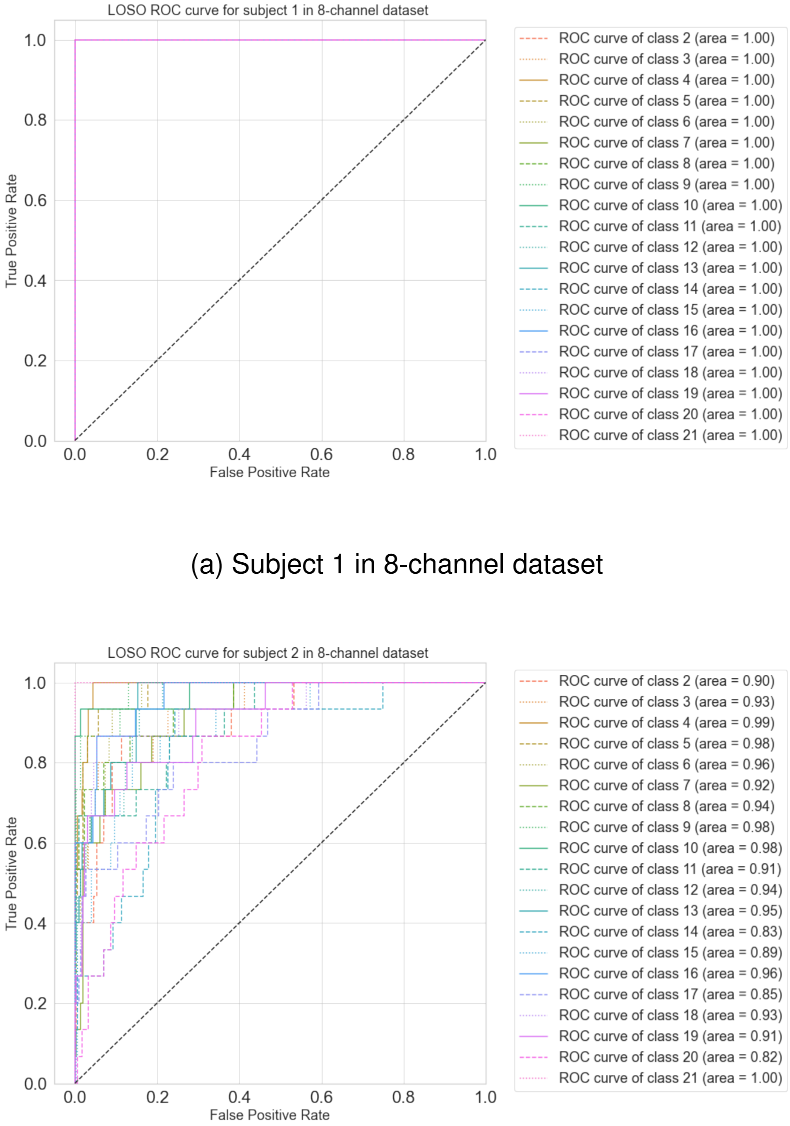
A.1.5 ROC Curve



(a) Subject 1 in 8-channel dataset

(b) Subject 2 in 8-channel dataset

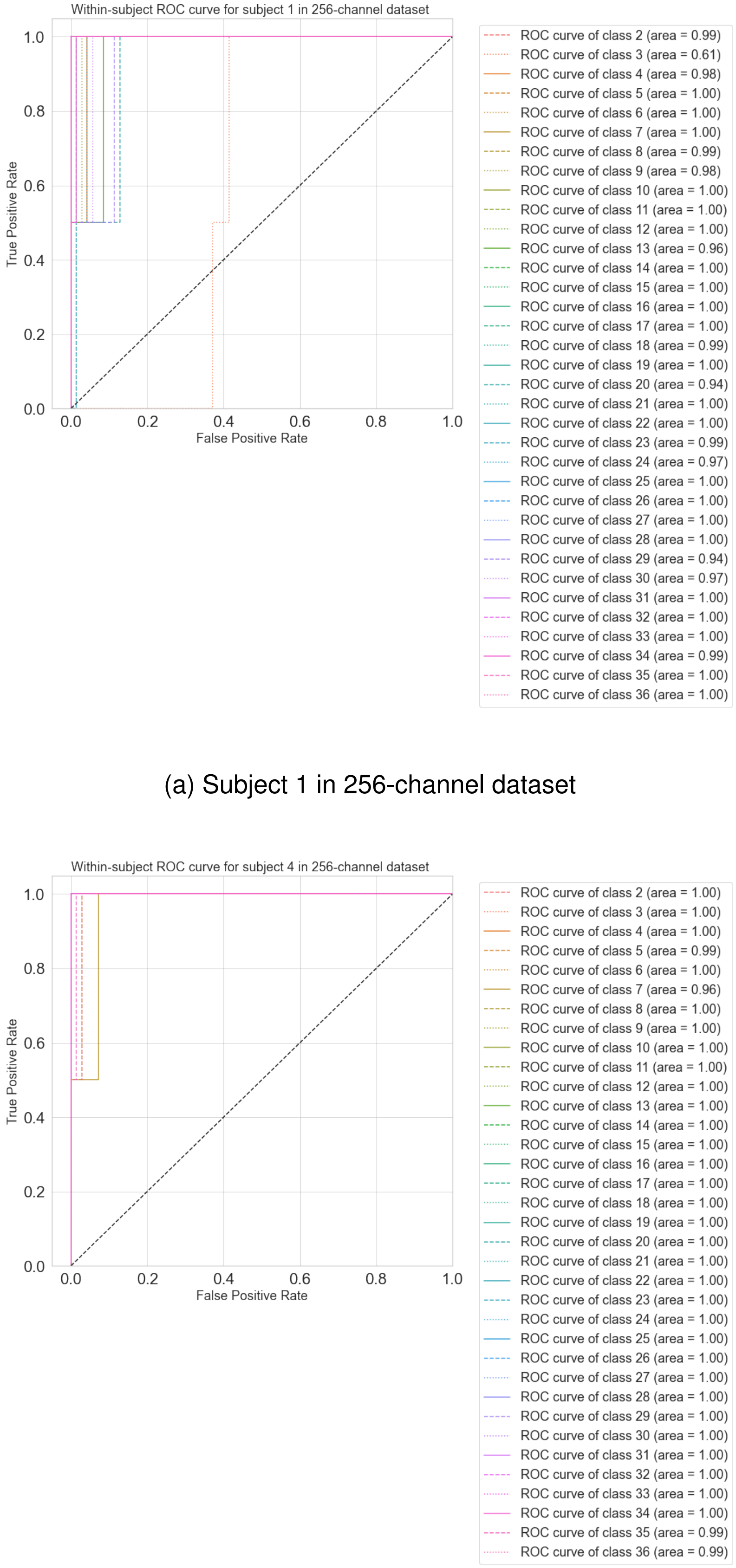
Figure A.27: Within-subject ROC-curves on 8-channel dataset



(a) Subject 1 in 8-channel dataset

(b) Subject 2 in 8-channel dataset

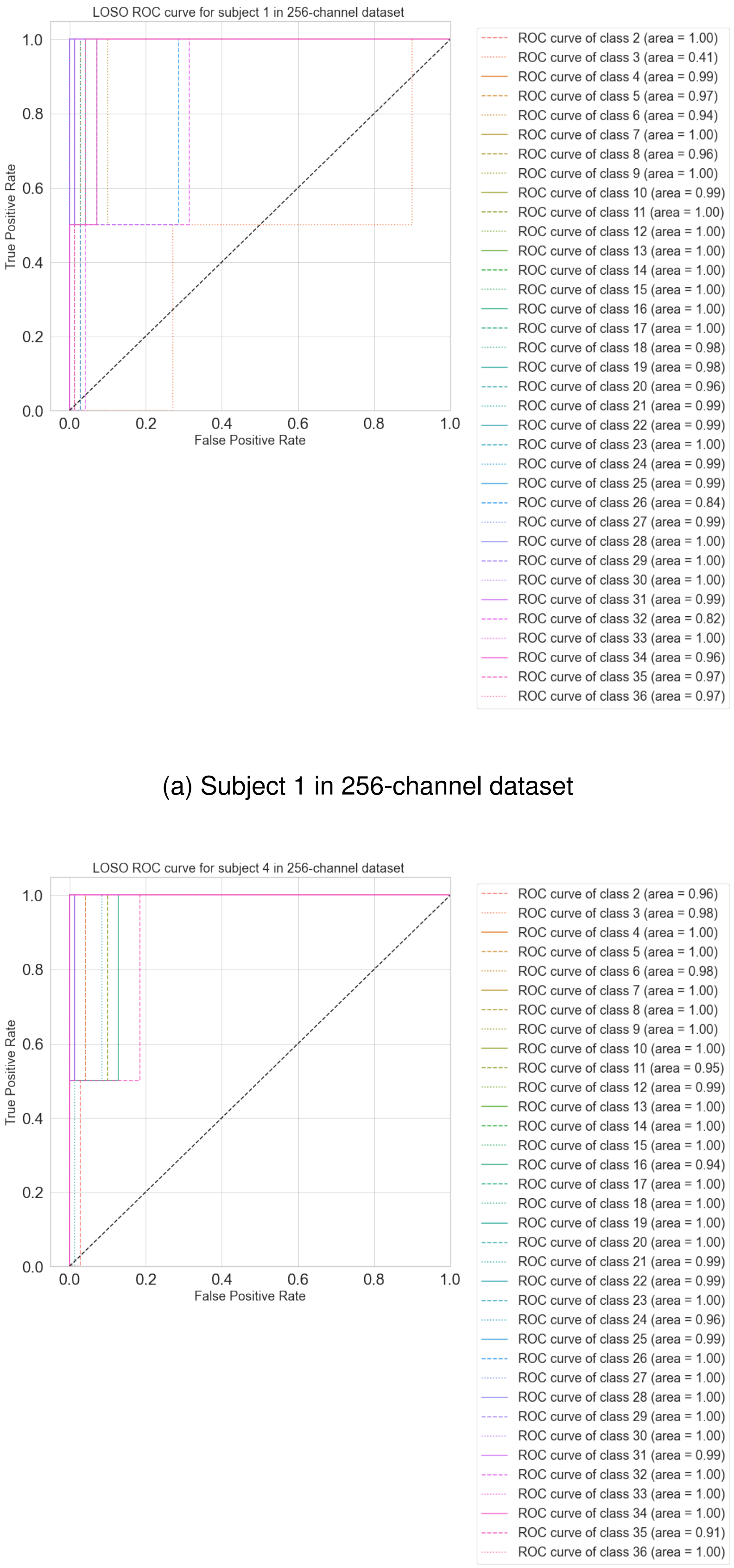
Figure A.28: LOSO ROC-curves on 8-channel dataset



(a) Subject 1 in 256-channel dataset

(b) Subject 4 in 256-channel dataset

Figure A.29: Within-subject ROC-curves on 256-channel dataset



(a) Subject 1 in 256-channel dataset

(b) Subject 4 in 256-channel dataset

Figure A.30: LOSO ROC-curves on 256-channel dataset

A.2 Comparison with other models (Results)

A.2.1 Accuracy

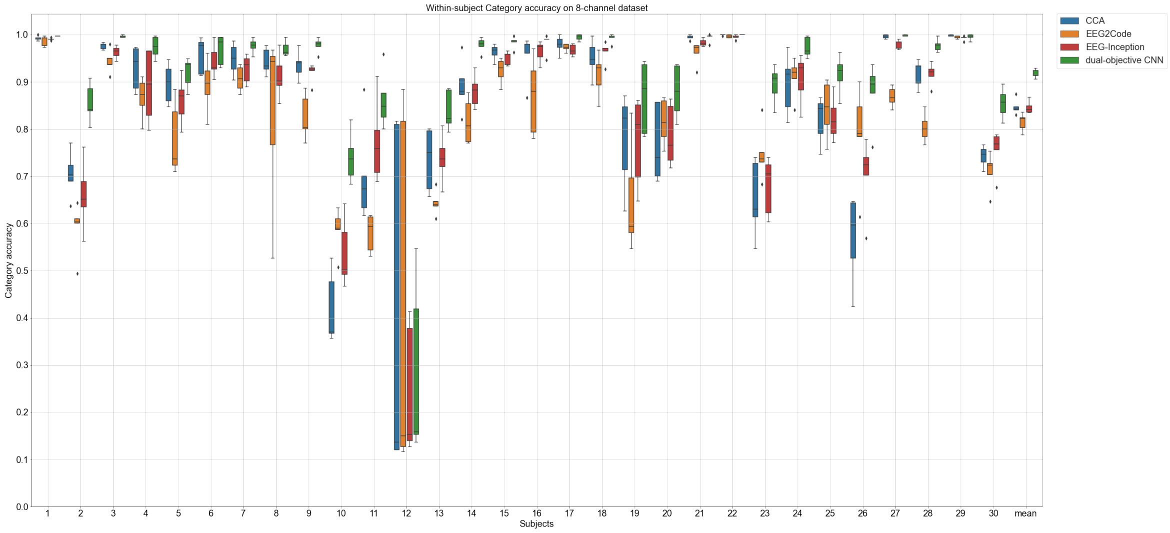


Figure A.31: Within-subject category accuracy of various models on 8-channel dataset

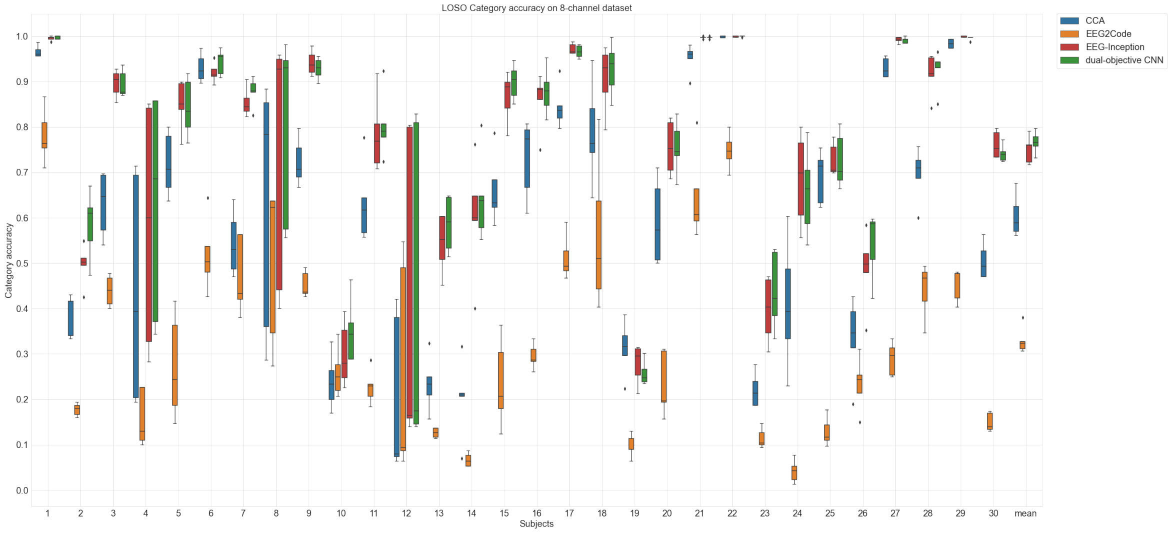


Figure A.32: LOSO category accuracy of various models on 8-channel dataset

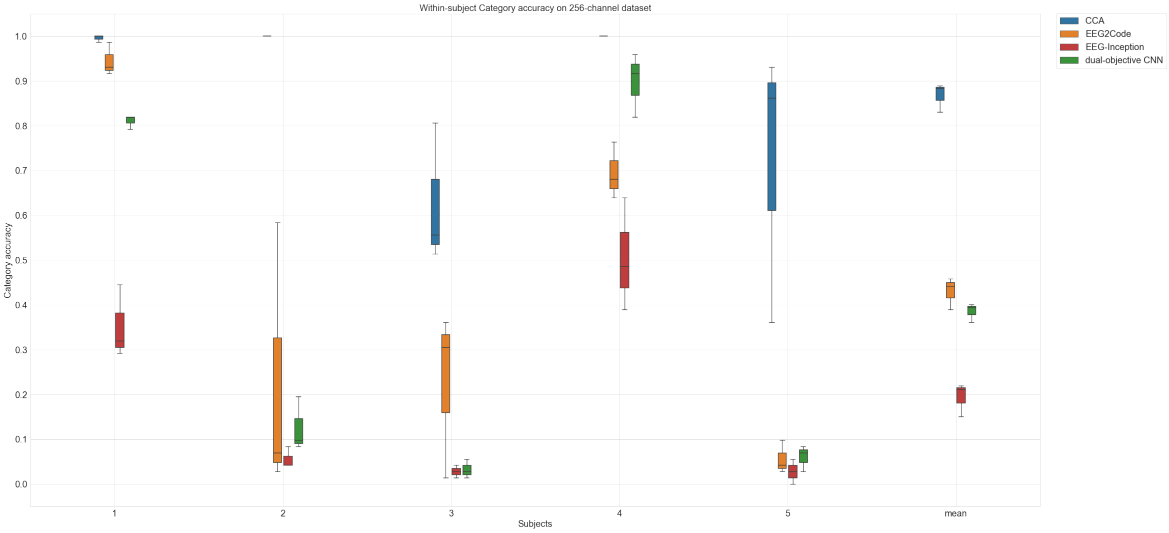


Figure A.33: Within-subject category accuracy of various models on 256-channel dataset

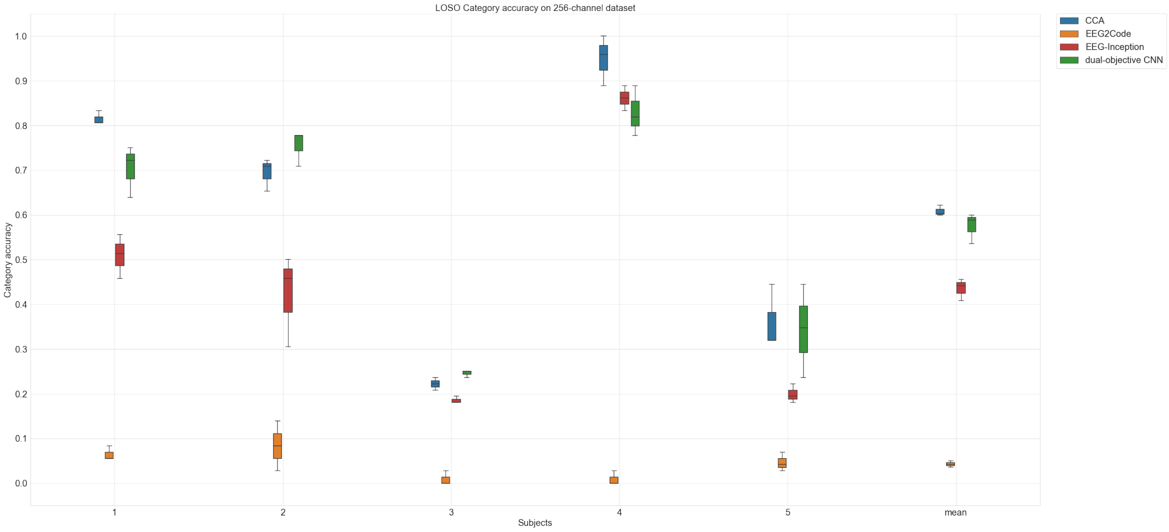


Figure A.34: LOSO category accuracy of various models on 256-channel dataset

A.2.2 Information transfer rate (ITR)

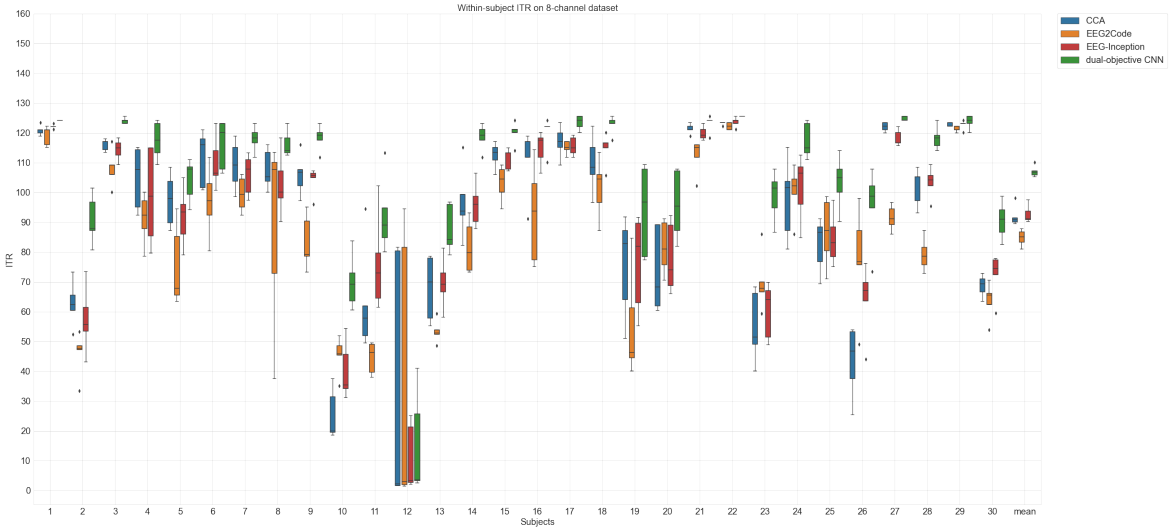


Figure A.35: Within-subject ITR of various models on 8-channel dataset

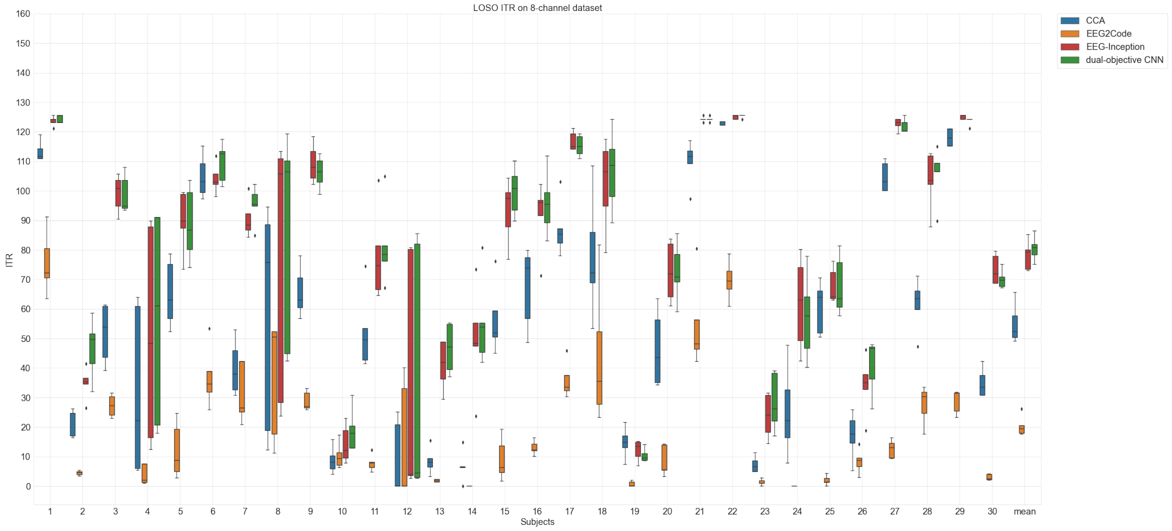


Figure A.36: LOSO ITR of various models on 8-channel dataset

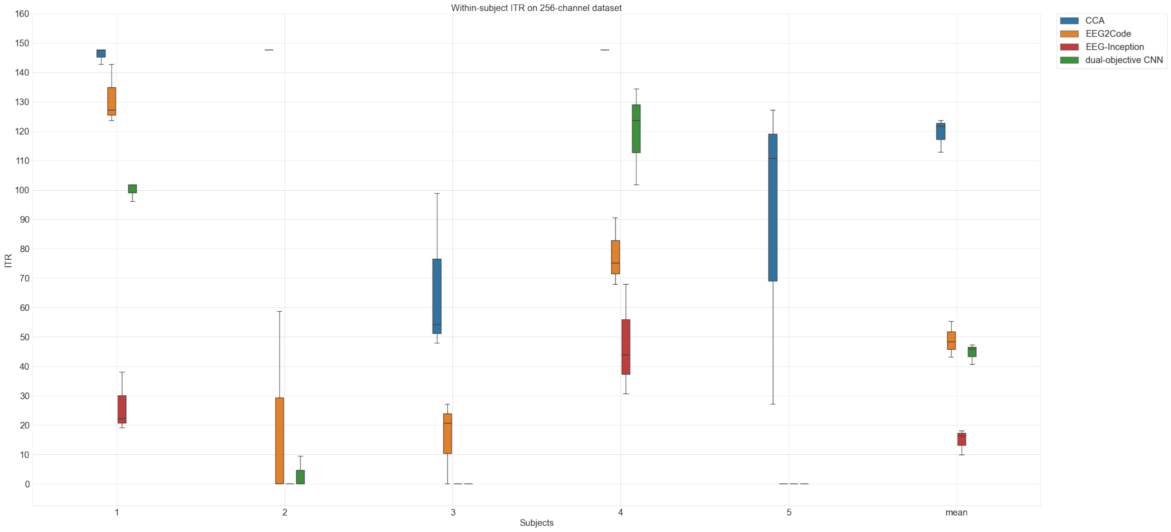


Figure A.37: Within-subject ITR of various models on 256-channel dataset

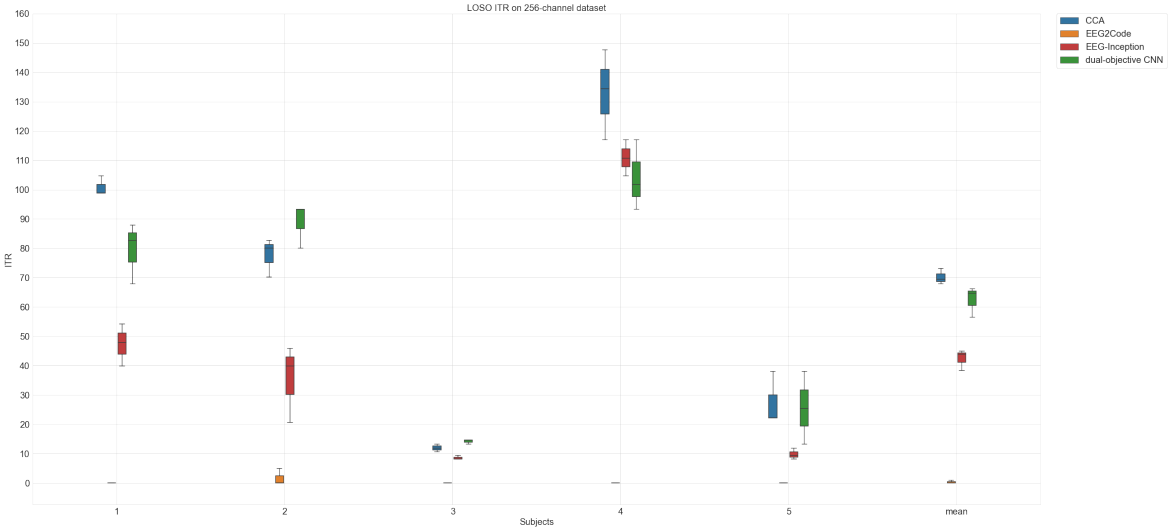


Figure A.38: LOSO ITR of various models on 256-channel dataset

A.2.3 Performance over time-steps

A.2.3.1 CCA

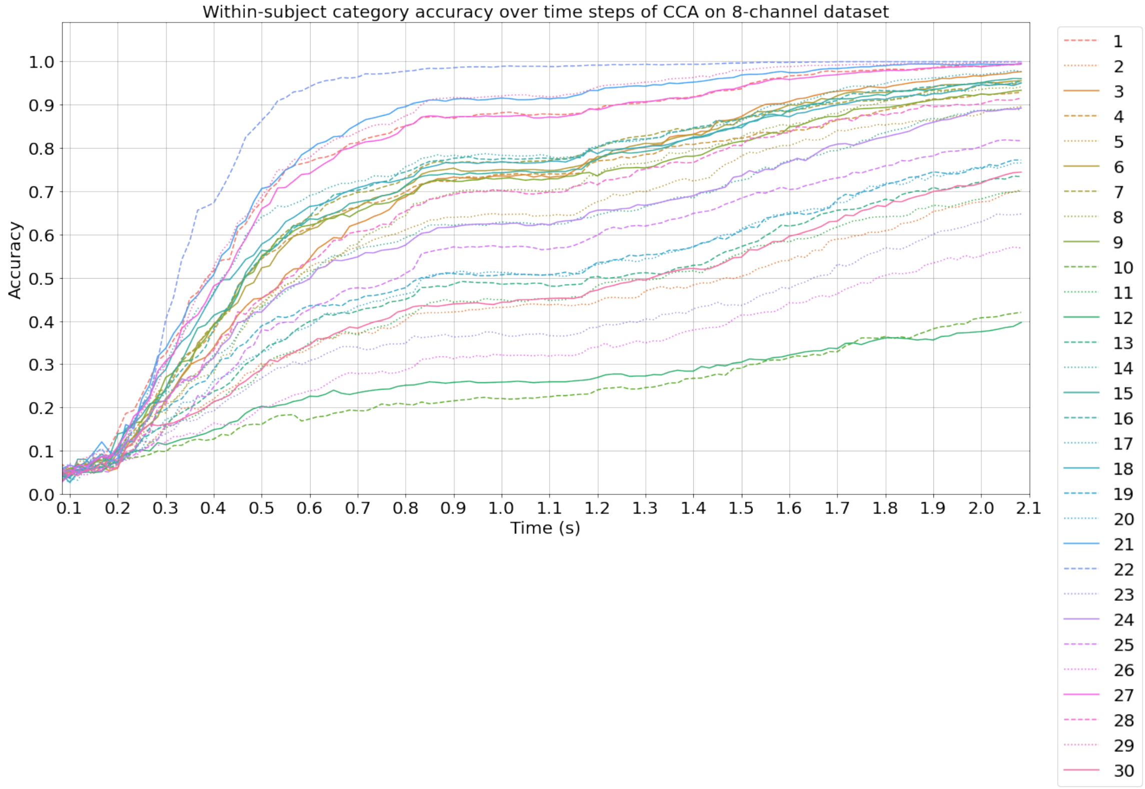


Figure A.39: Within-subject category accuracy over time-steps of CCA on 8-channel dataset

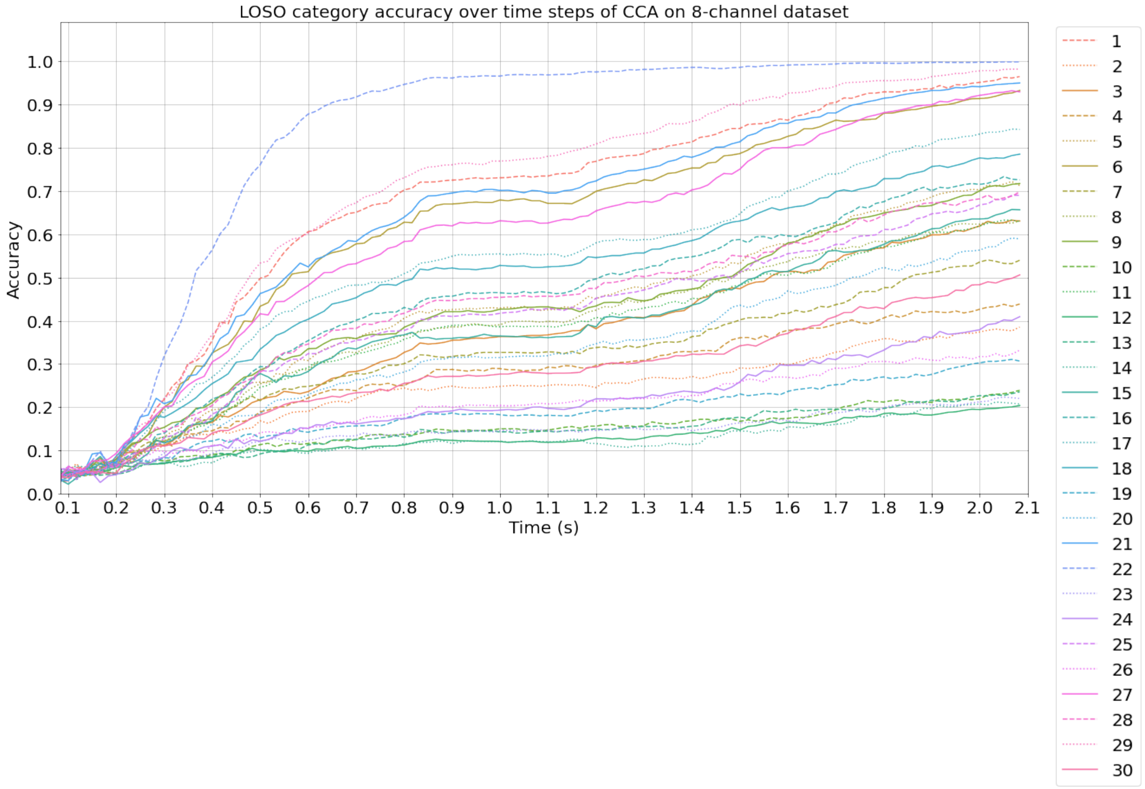


Figure A.40: LOSO category accuracy over time-steps of CCA on 8-channel dataset

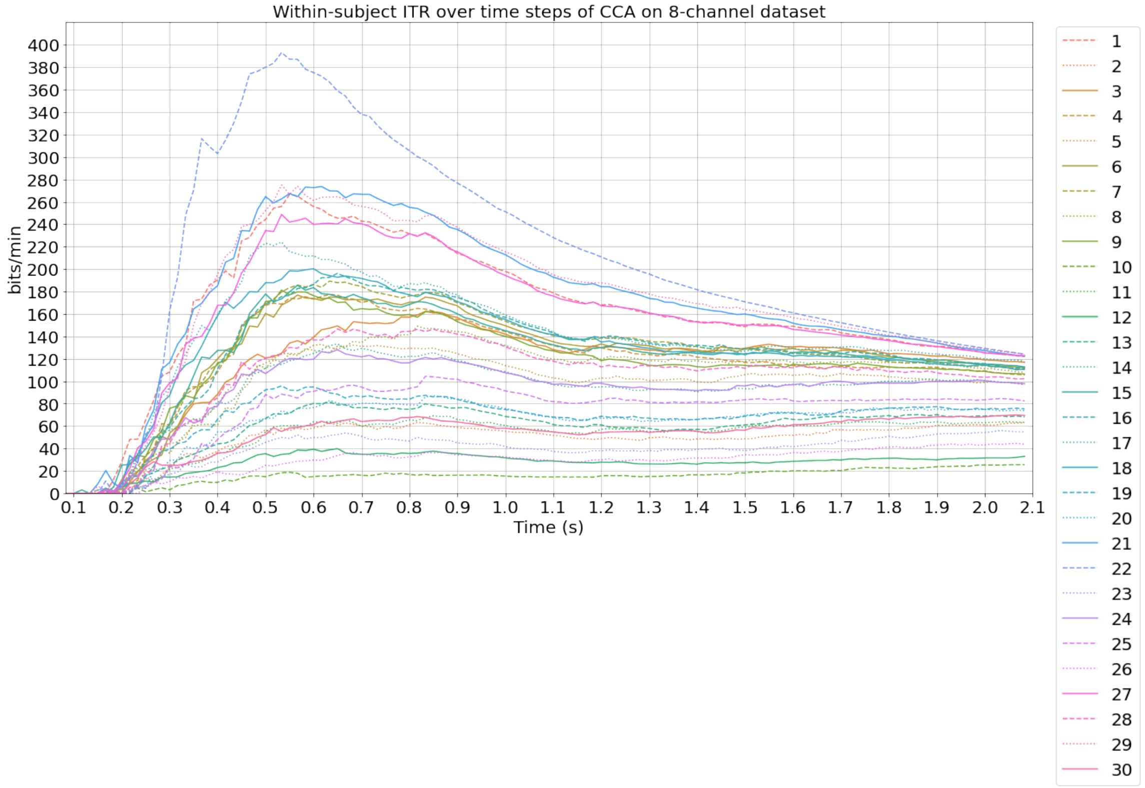


Figure A.41: Within-subject ITR over time-steps of CCA on 8-channel dataset

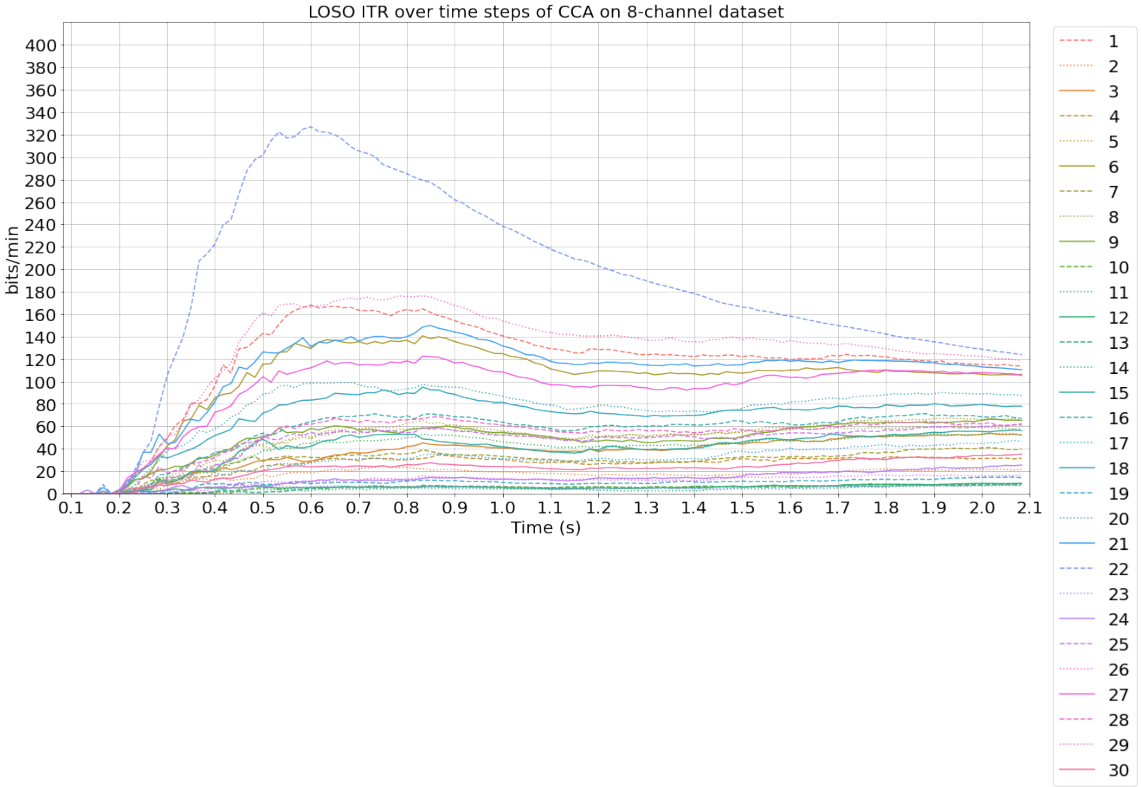


Figure A.42: LOSO ITR over time-steps of CCA on 8-channel dataset

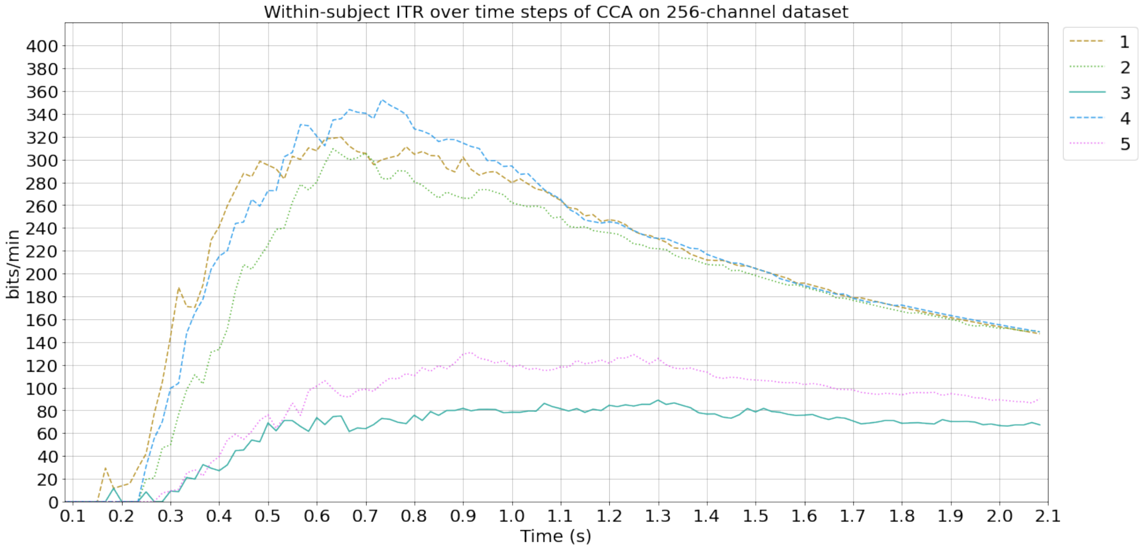


Figure A.43: Within-subject category accuracy over time-steps of CCA on 256-channel dataset

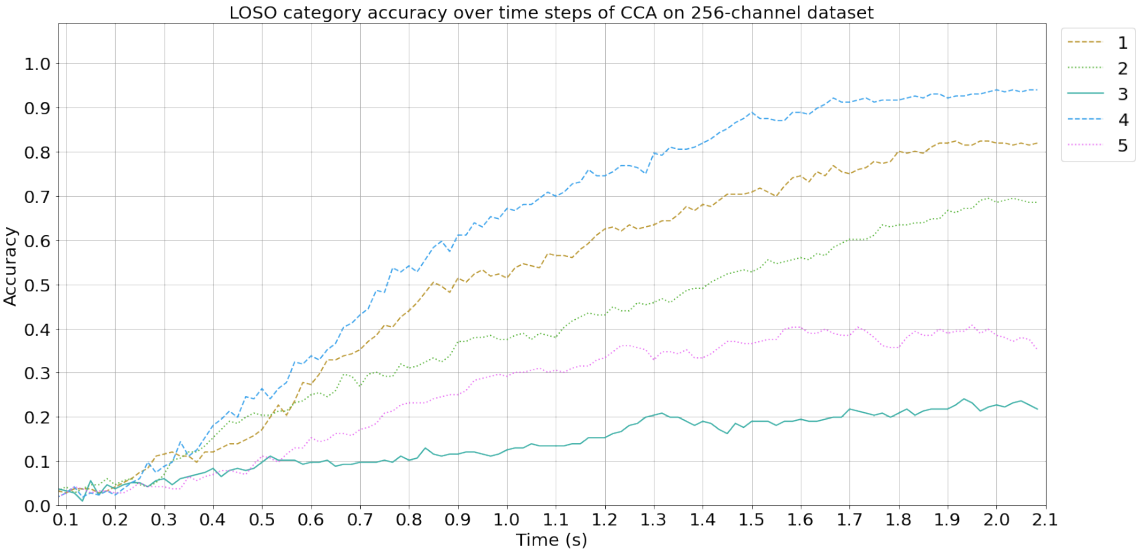


Figure A.44: LOSO category accuracy over time-steps of CCA on 256-channel dataset

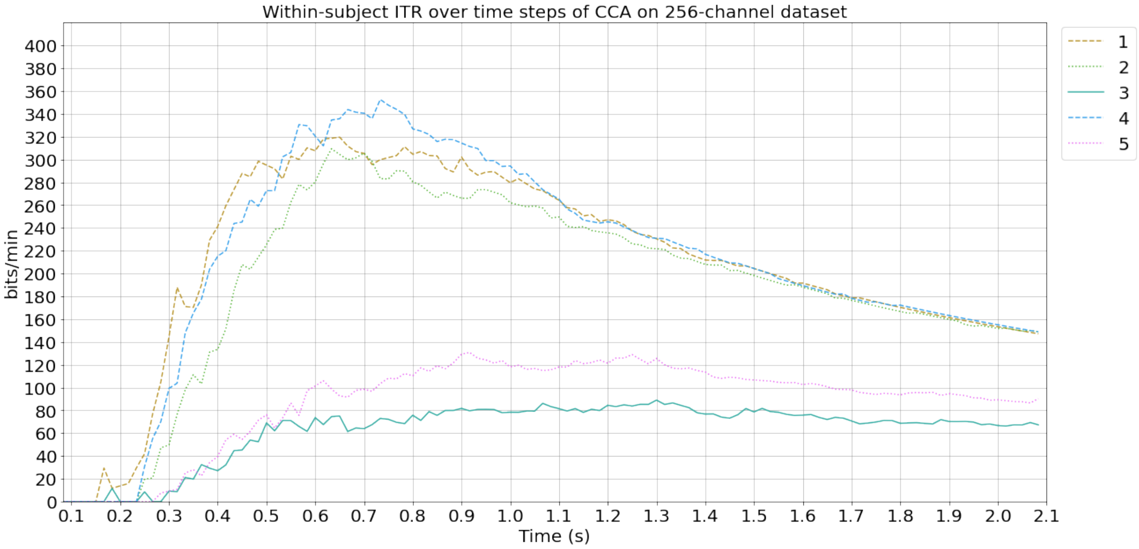


Figure A.45: Within-subject ITR over time-steps of CCA on 256-channel dataset

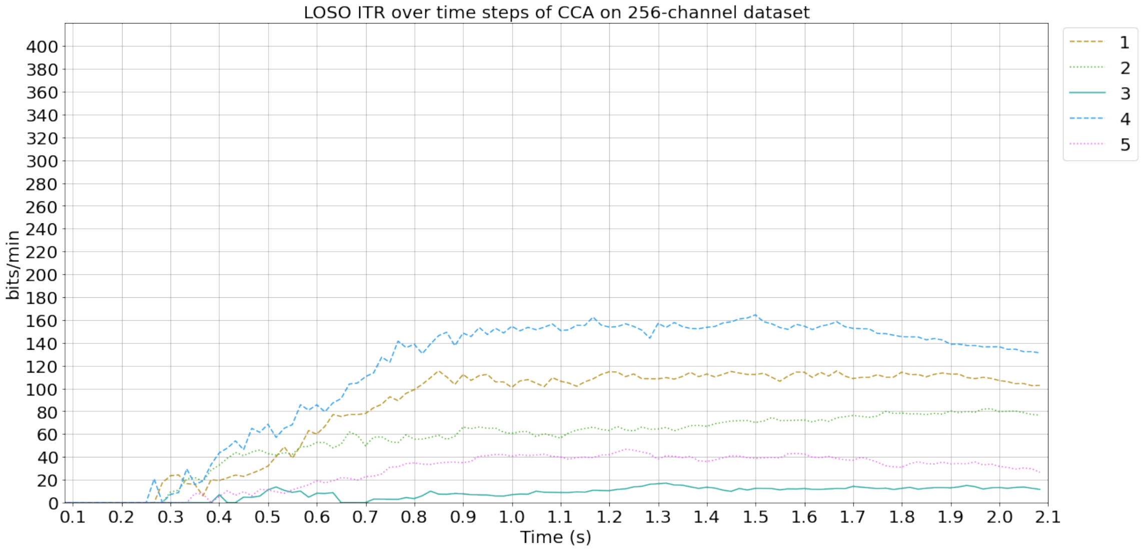


Figure A.46: LOSO ITR over time-steps of CCA on 256-channel dataset

A.2.3.2 EEG2Code

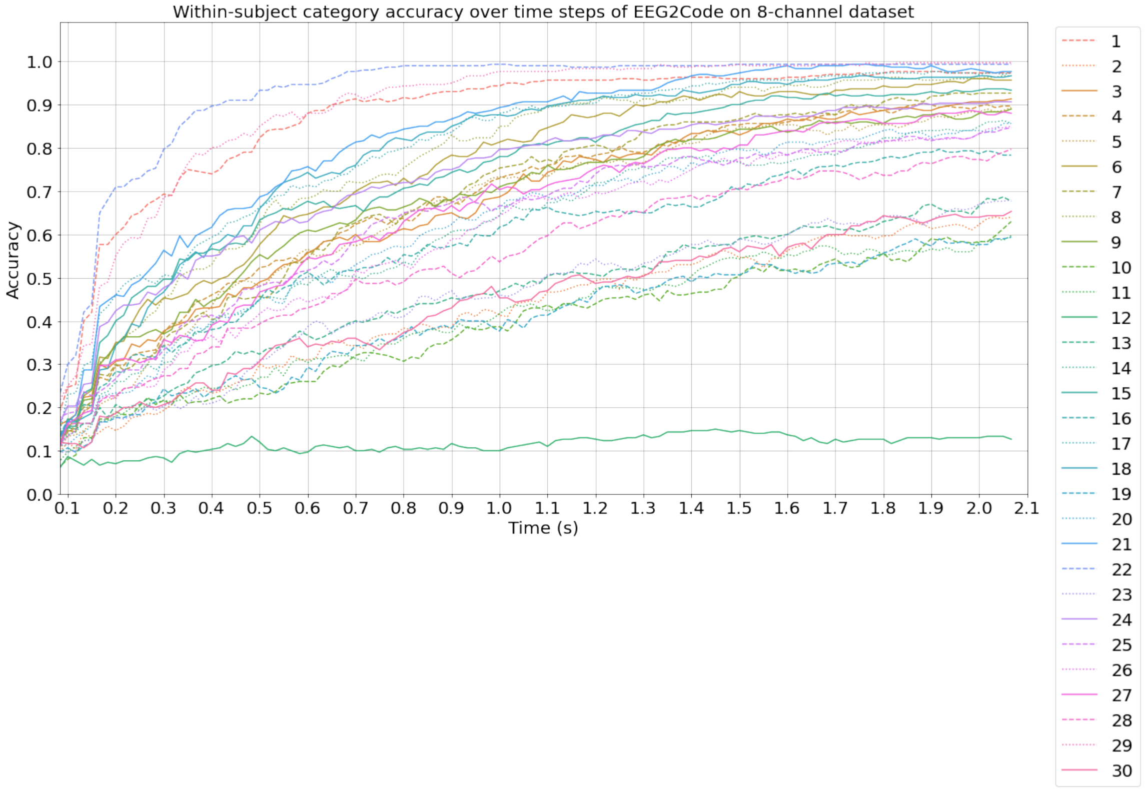


Figure A.47: Within-subject category accuracy over time-steps of EEG2Code on 8-channel dataset

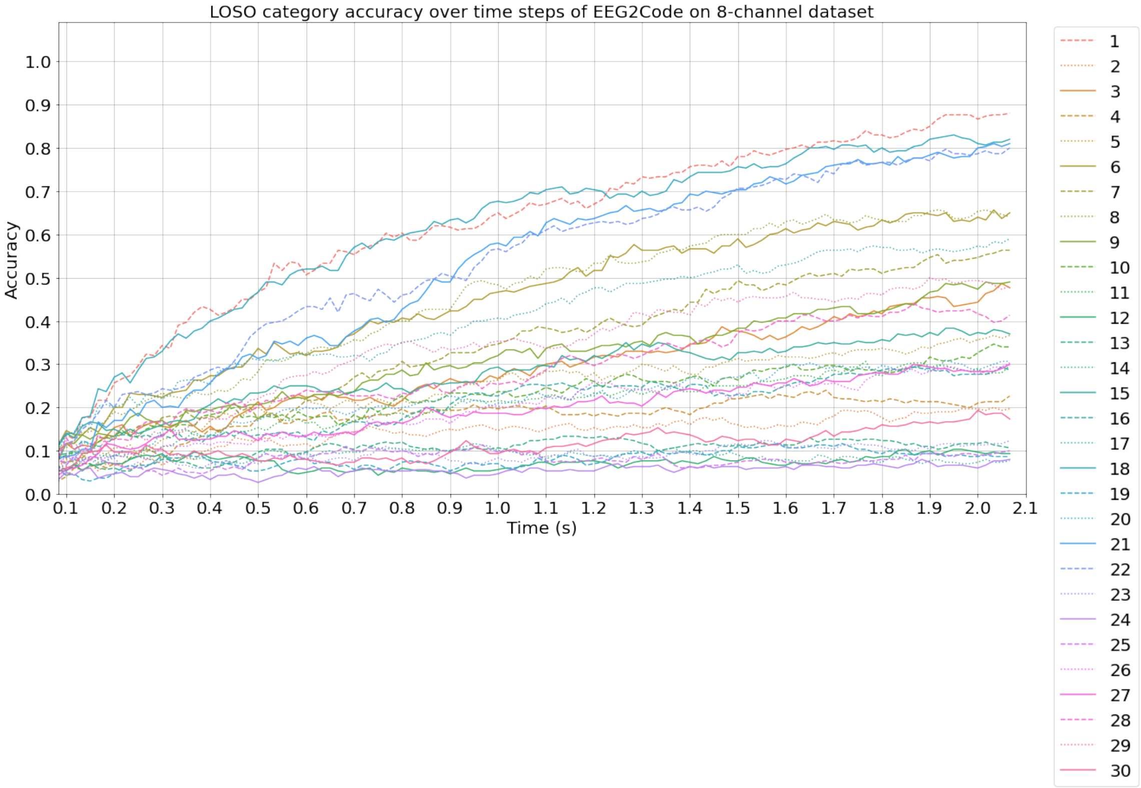


Figure A.48: LOSO category accuracy over time-steps of EEG2Code on 8-channel dataset

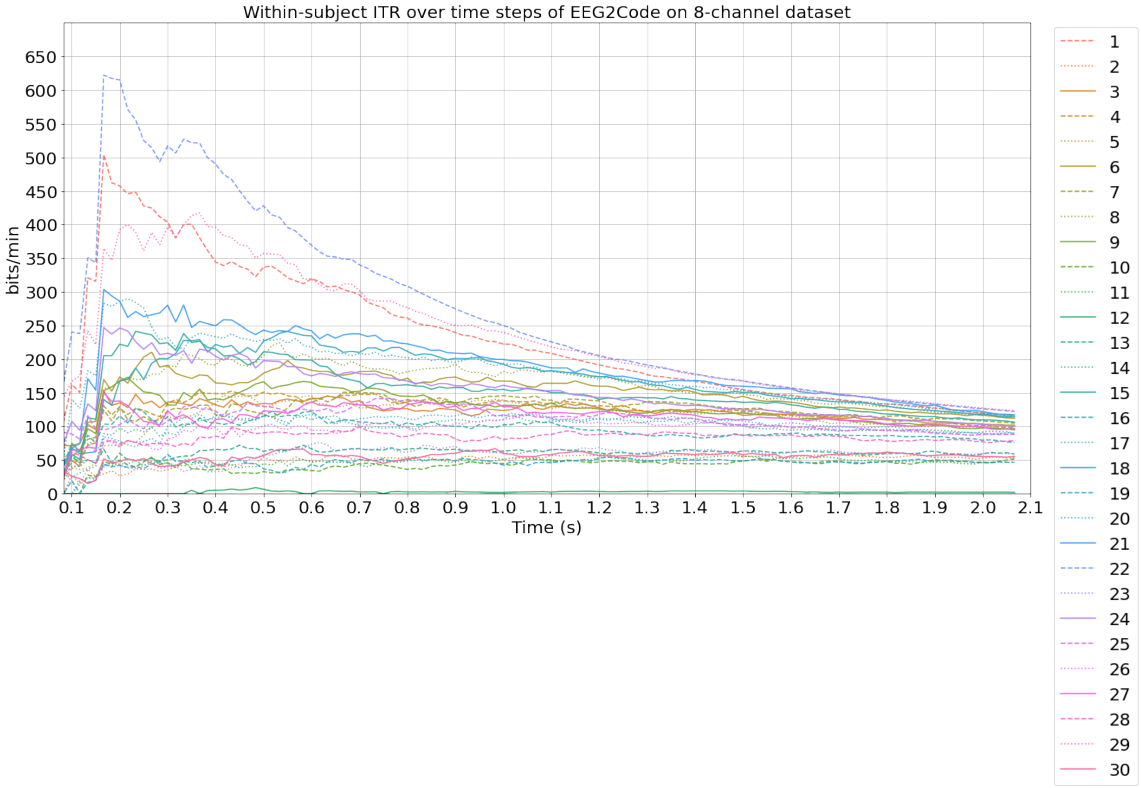


Figure A.49: Within-subject ITR over time-steps of EEG2Code on 8-channel dataset

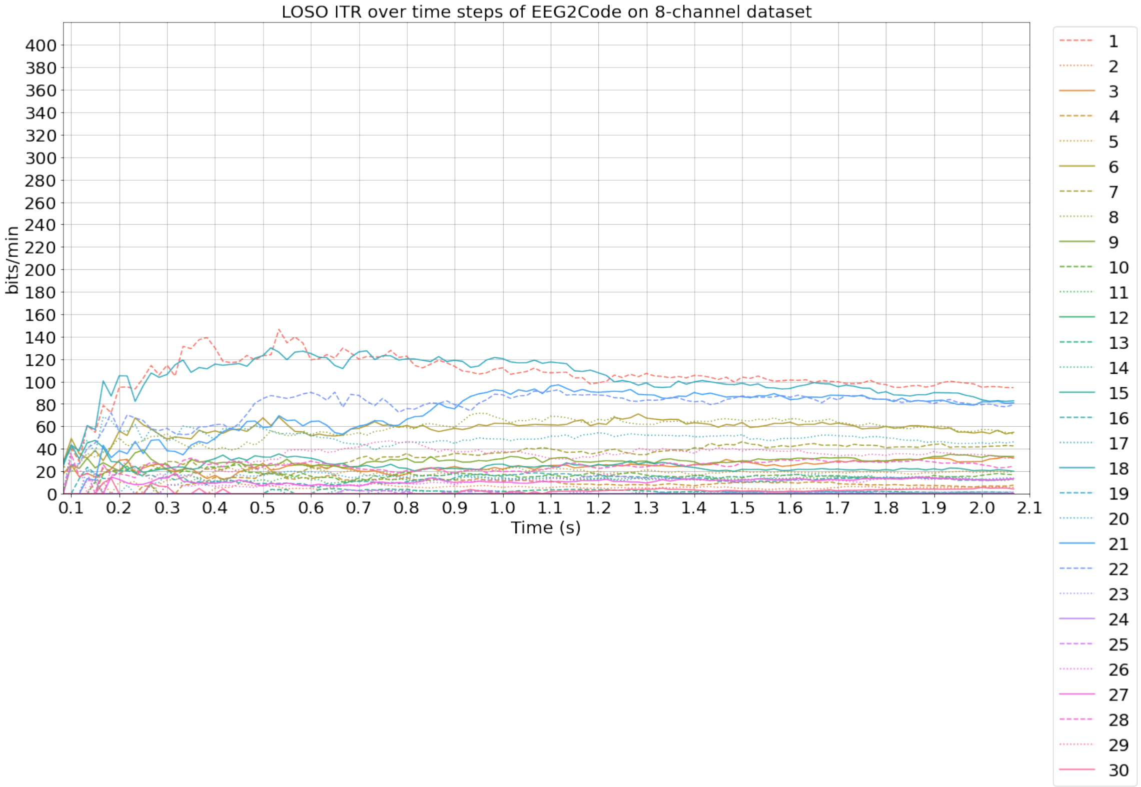


Figure A.50: LOSO ITR over time-steps of EEG2Code on 8-channel dataset

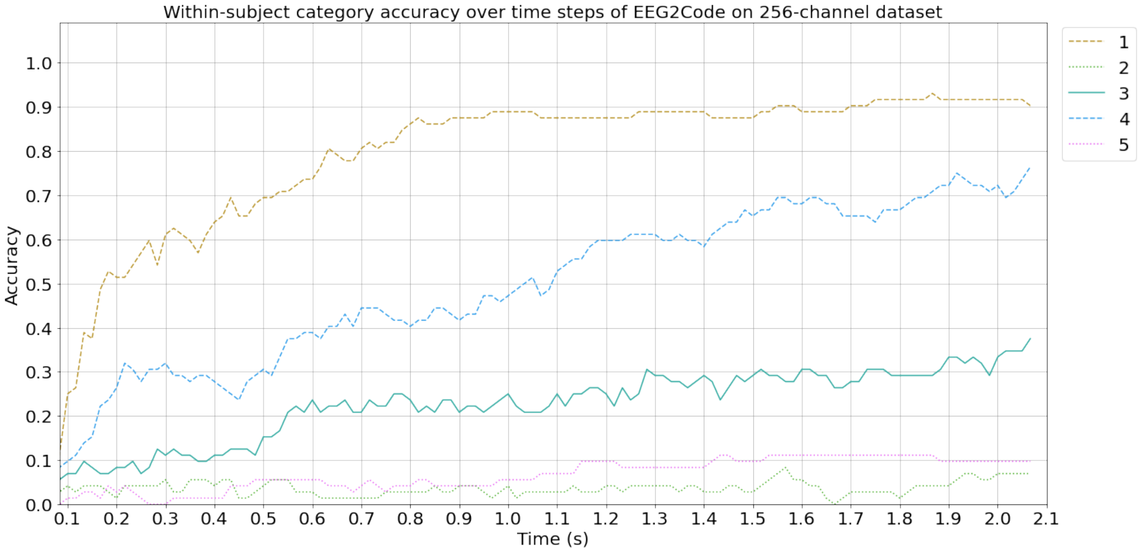


Figure A.51: Within-subject category accuracy over time-steps of EEG2Code on 256-channel dataset

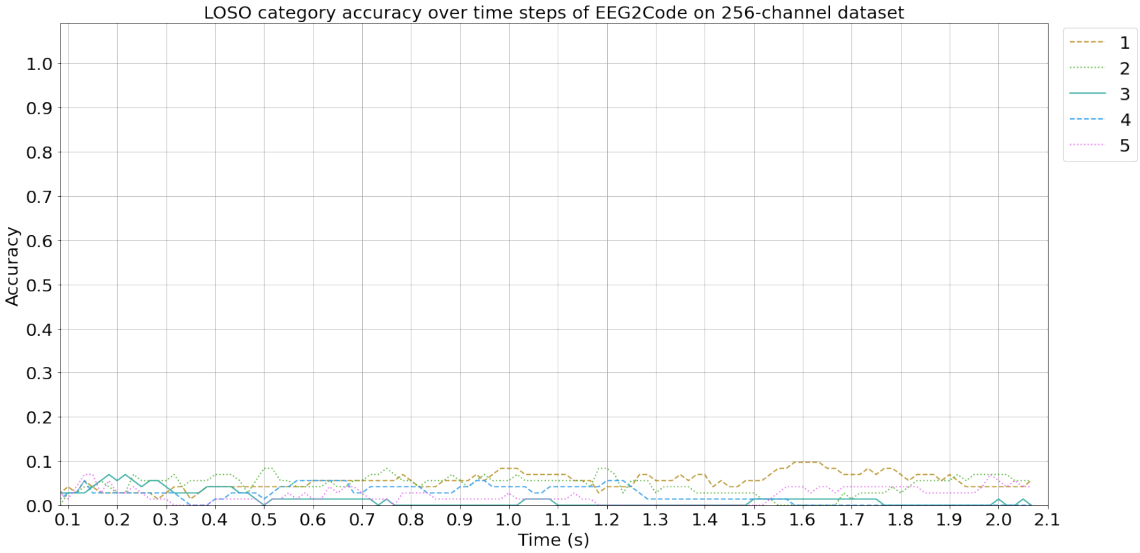


Figure A.52: LOSO category accuracy over time-steps of EEG2Code on 256-channel dataset

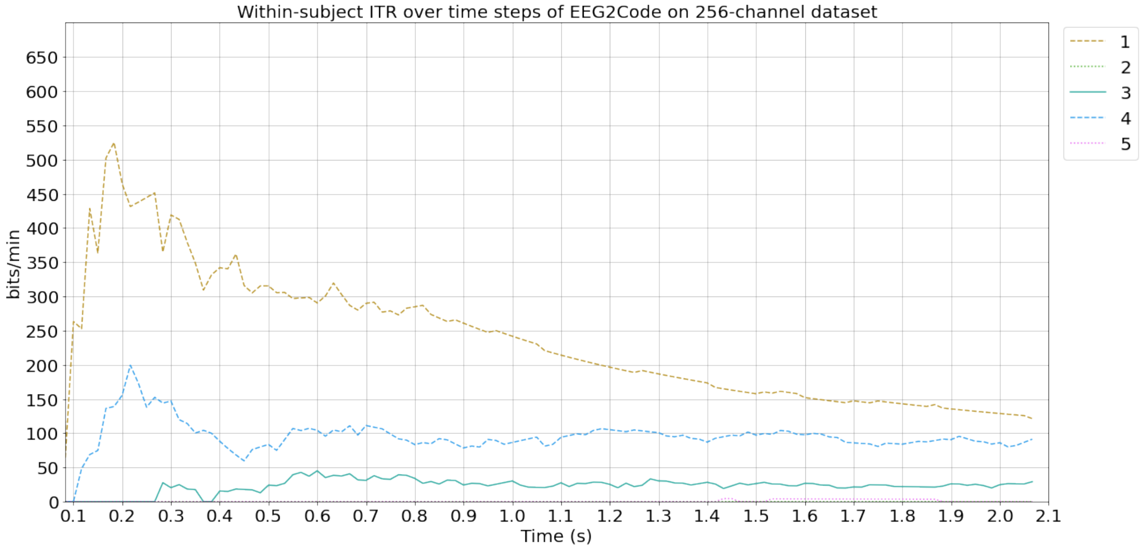


Figure A.53: Within-subject ITR over time-steps of EEG2Code on 256-channel dataset

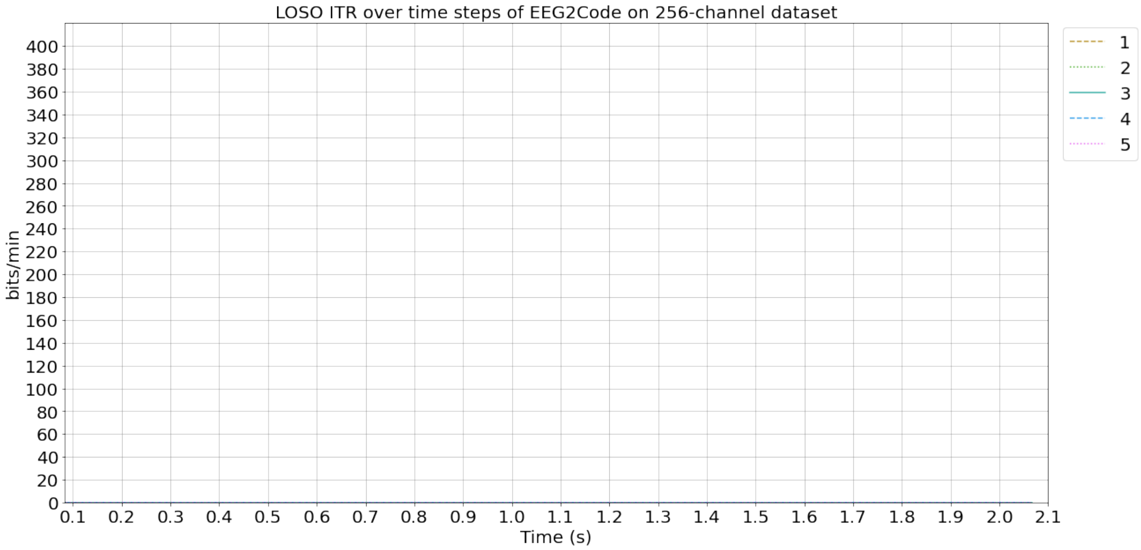


Figure A.54: LOSO ITR over time-steps of EEG2Code on 256-channel dataset

A.2.3.3 EEG-Inception

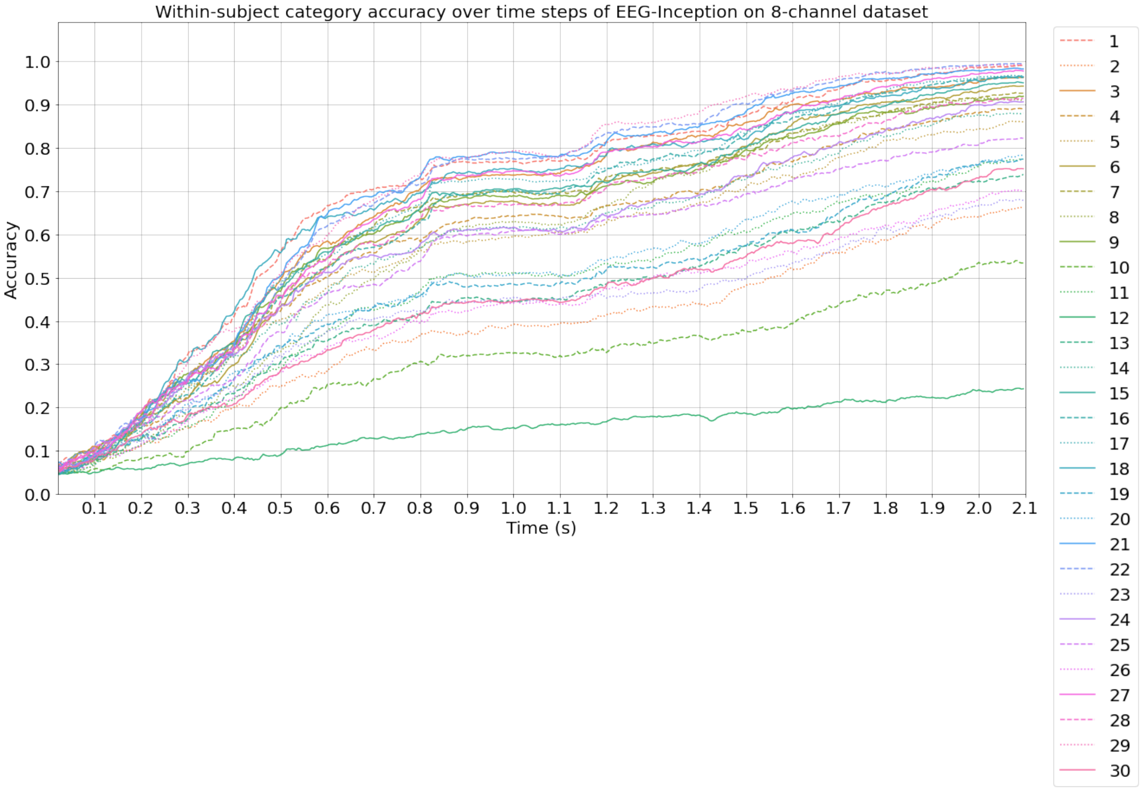


Figure A.55: Within-subject category accuracy over time-steps of EEG-Inception on 8-channel dataset

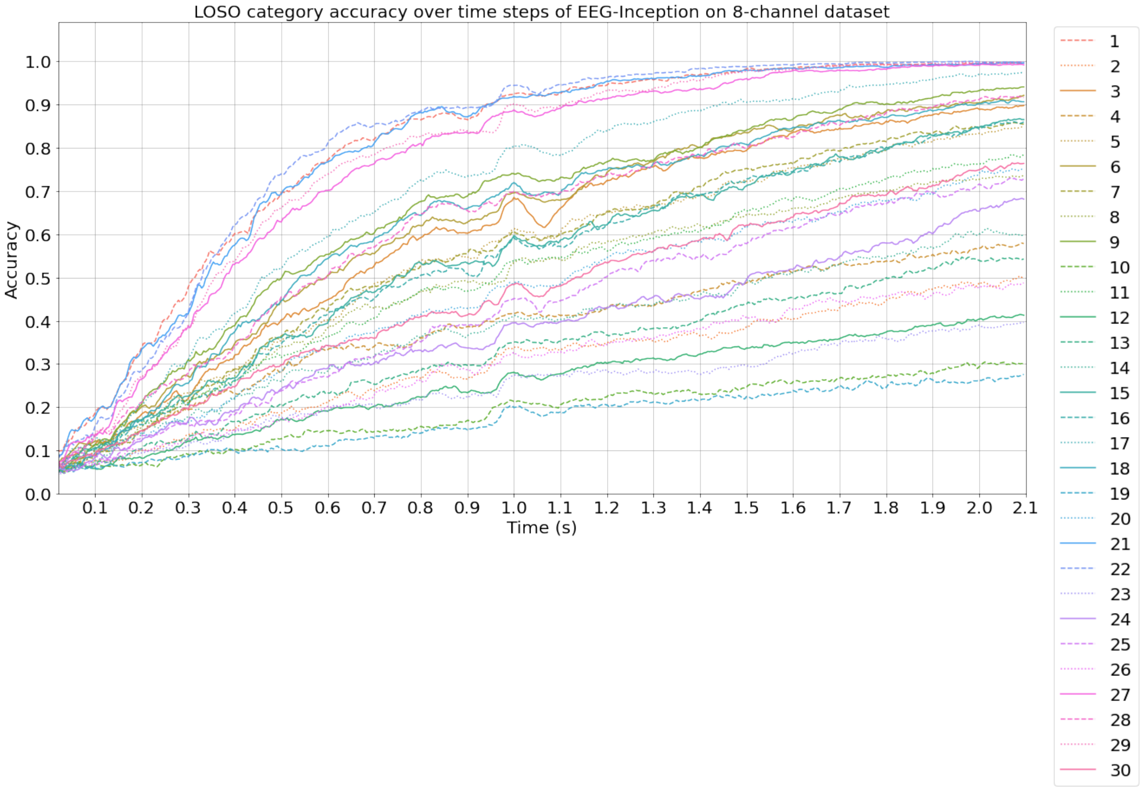


Figure A.56: LOSO category accuracy over time-steps of EEG-Inception on 8-channel dataset

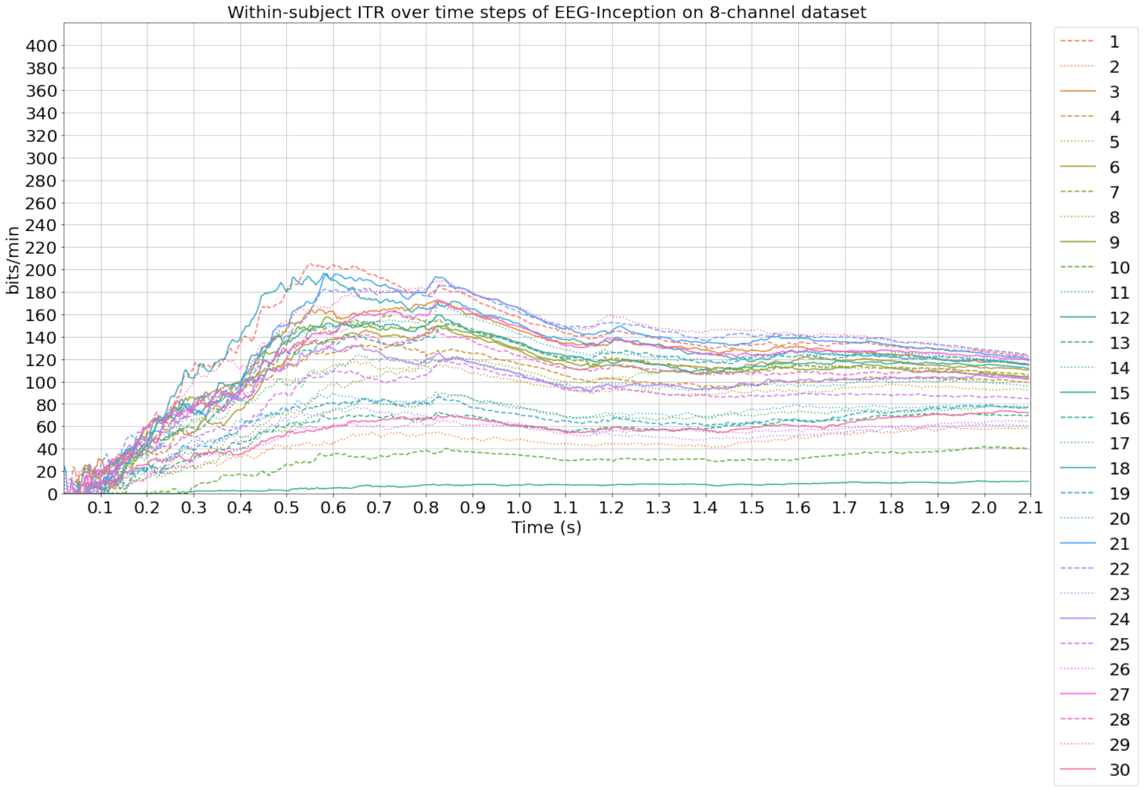


Figure A.57: Within-subject ITR over time-steps of EEG-Inception on 8-channel dataset

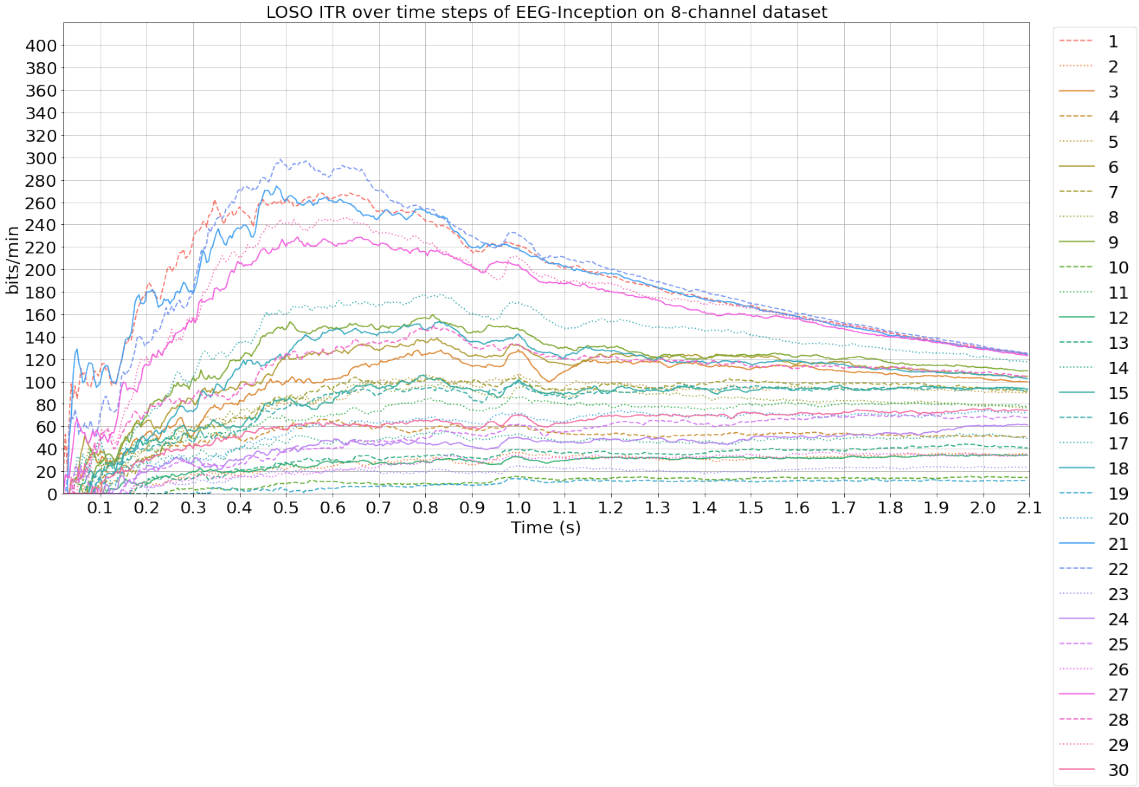


Figure A.58: LOSO ITR over time-steps of EEG-Inception on 8-channel dataset

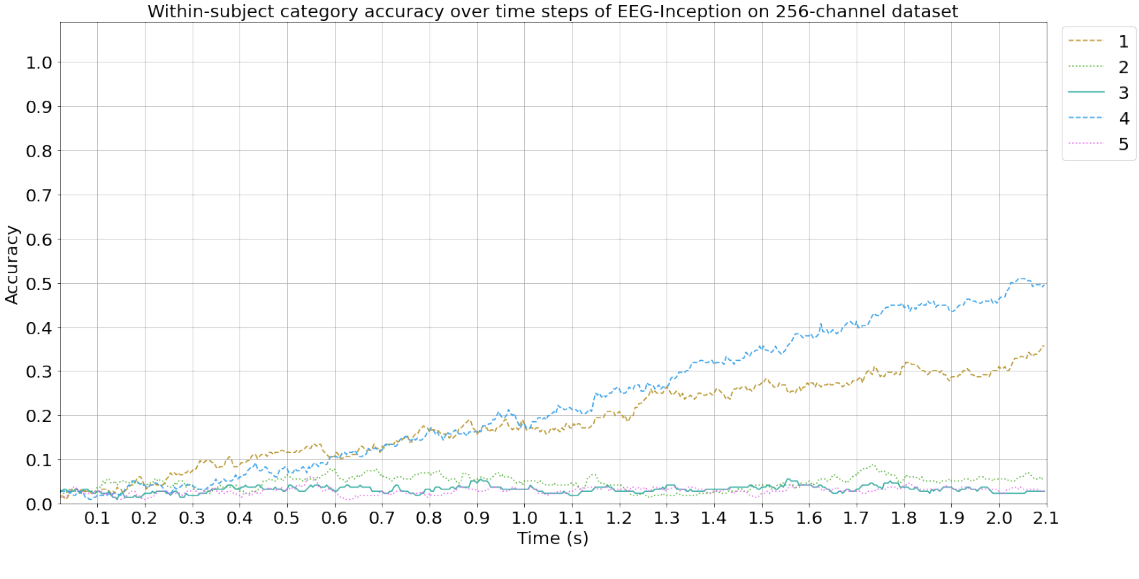


Figure A.59: Within-subject category accuracy over time-steps of EEG-Inception on 256-channel dataset

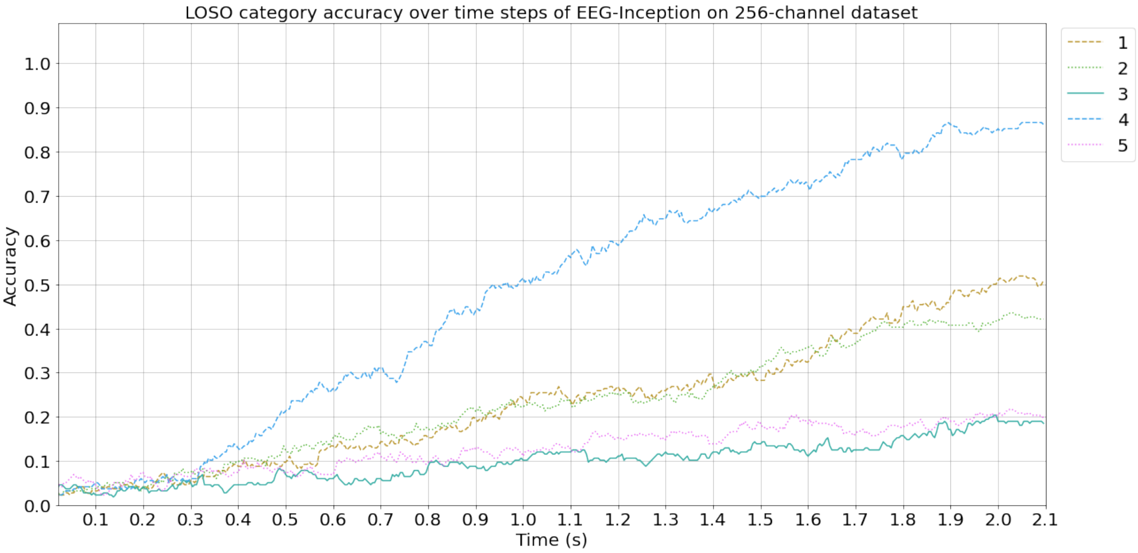


Figure A.60: LOSO category accuracy over time-steps of EEG-Inception on 256-channel dataset

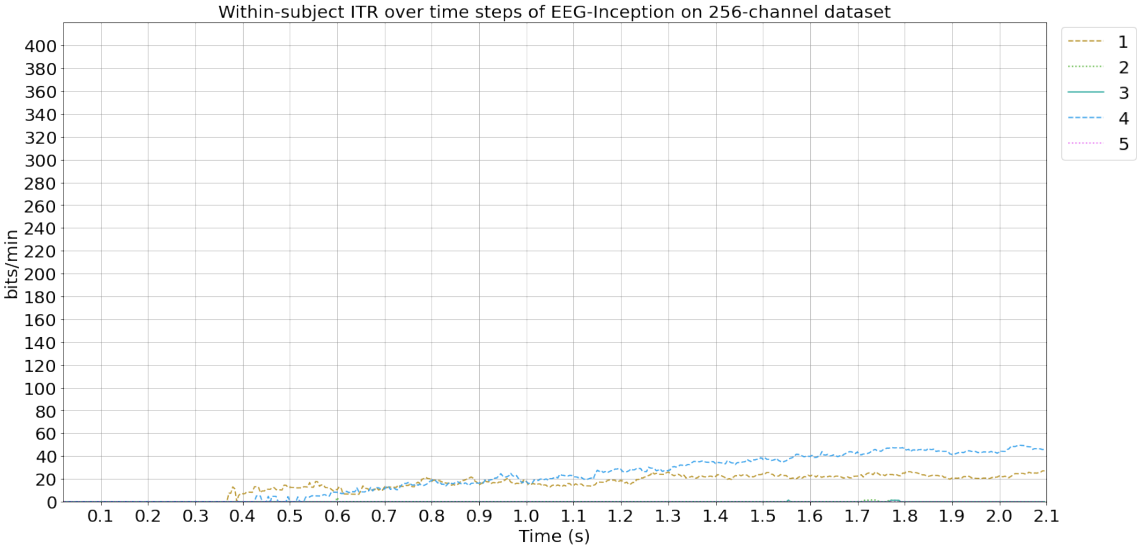


Figure A.61: Within-subject ITR over time-steps of EEG-Inception on 256-channel dataset

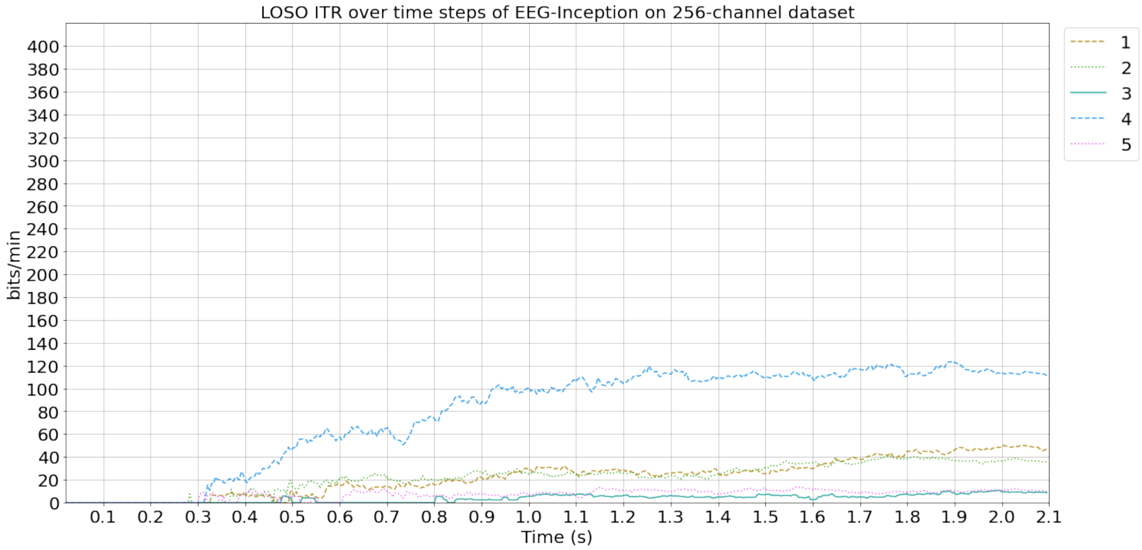


Figure A.62: LOSO ITR over time-steps of EEG-Inception on 256-channel dataset

Bibliography

[1] Dennis J. McFarland and Jonathan R. Wolpaw. Brain-computer interfaces for communication and control. *Commun. ACM*, 54(5):6066, may 2011. ISSN 0001-0782. doi: 10.1145/1941487.1941506. URL https://doi.org/10.1145/1941487.1941506. 1

[2] Guangyu Bin, Xiaorong Gao, Yijun Wang, Bo Hong, and Shangkai Gao. Vep-based brain-computer interfaces: time, frequency, and code modulations [research frontier]. *IEEE Computational Intelligence Magazine*, 4(4):22–26, 2009. doi: 10.1109/MCI.2009.934562. 1, 2.3, 2.4

[3] Sebastian Nagel and Martin Spüler. Worlds fastest brain-computer interface: Combining eeg2code with deep learning. *PLOS ONE*, 14(9):1–15, 09 2019. doi: 10.1371/journal.pone.0221909. URL https://doi.org/10.1371/journal.pone.0221909. 1, 3.3, 3.5, 3.6, 5.3.1

[4] Aya Rezeika, Mihaly Benda, Piotr Stawicki, Felix Gembler, Abdul Saboor, and Ivan Volosyak. Braincomputer interface spellers: A review. *Brain Sciences*, 8, 2018. 1

[5] Jordy Thielen, Pieter Marsman, Jason Farquhar, and Peter Desain. From full calibration to zero training for a code-modulated visual evoked potentials brain computer interface. *Journal of Neural Engineering*, 18, 03 2021. doi: 10.1088/1741-2552/abecef. 1, 2.2, 2.4, 3.2, 3.5, 3.6, 4.1, 5.1.2, 5.1.3

[6] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458. 1

[7] Vinay Jayaram, Morteza Alamgir, Yasemin Altun, Bernhard Scholkopf, and Moritz Grosse-Wentrup. Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, 11:20–31, 02 2016. doi: 10.1109/MCI.2015.2501545. 1

[8] Eduardo López-Larraz, Andrea Sarasola-Sanz, Nerea Irastorza-Landa, Niels Birbaumer, and Ander Ramos-Murguialday. Brain-machine interfaces for rehabilitation in stroke: A review. *NeuroRehabilitation*, 43 1:77–97, 2018. 2.1

[9] Javier Gómez-Pilar, Rebeca Corralejo, Luis F. Nicolás-Alonso, Daniel Álvarez, and Roberto Hornero. Neurofeedback training with a motor imagery-based bci: neurocognitive improvements and eeg changes in the elderly. *Medical & Biological Engineering & Computing*, 54:1655–1666, 2016. 2.1

[10] P Aricò, G Borghini, G Di Flumeri, N Sciaraffa, and F Babiloni. Passive bci beyond the lab: current trends and future directions. *Physiological Measurement*, 39(8):08TR02, August 2018. ISSN 0967-3334. doi: 10.1088/1361-6579/aad57e. 2.1

[11] C. Klaes. *Invasive Brain-Computer Interfaces and Neural Recordings From Humans*, pages 527–539. 01 2019. doi: 10.1016/B978-0-12-812028-6.00028-8. 2.1

[12] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *Sensors (Basel, Switzerland)*, 12, 2012. doi: 10.3390/s120201211. 2.2

[13] Erich E. Sutter. The brain response interface: communication through visually-induced electrical brain responses. *Journal of Microcomputer Applications*, 15 (1):31–45, 1992. ISSN 0745-7138. doi: https://doi.org/10.1016/0745-7138(92) 90045-7. URL https://www.sciencedirect.com/science/article/pii/0745713892900457. Special Issue on Computers for Handicapped People. 2.3

[14] Qingguo Wei, Siwei Feng, and Zongwu Lu. Stimulus specificity of brain-computer interfaces based on code modulation visual evoked potentials. *PLOS ONE*, 11(5):1–17, 05 2016. doi: 10.1371/journal.pone.0156416. URL https://doi.org/10.1371/journal.pone.0156416. 2.3

[15] Víctor Martínez-Cagigal, Jordy Thielen, Eduardo Santamaría-Vázquez, Sergio Pérez-Velasco, Peter Desain, and Roberto Hornero. Brain–computer interfaces based on code-modulated visual evoked potentials (c-VEP): a literature review. *Journal of Neural Engineering*, 18(6):061002, nov 2021. doi: 10.1088/1741-2552/ac38cf. URL https://doi.org/10.1088/1741-2552/ac38cf. 2.3, 3.1, 3.3

[16] J. K. Holmes. *Spread Spectrum Systems for GNSS and Wireless Communications*. Artech House, 2007. 2.4

[17] Jonas Isaksen, Ali Mohebbi, and Sadasivan Puthusserypady. A comparative study of pseudorandom sequences used in a c-vep based bci for online wheelchair control. volume 2016, pages 1512–1515, 08 2016. doi: 10.1109/EMBC.2016.7590997. 2.4

[18] Muhammad Nabi Yasinzai and Yusuf Ziya Ider. New approach for designing cVEP BCI stimuli based on superposition of edge responses. *Biomedical Physics &amp Engineering Express*, 6(4):045018, jun 2020. doi: 10.1088/2057-1976/ab98e7. URL https://doi.org/10.1088%2F2057-1976%2Fab98e7. 2.4, 3.1, 3.2, 3.3, 3.6

[19] Sebastian Nagel, Martin Spüler, et al. Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed brain-computer interface. *PLOS ONE*, 13(10):1–16, 10 2018. doi: 10.1371/journal.pone.0206107. URL https://doi.org/10.1371/journal.pone.0206107. 2.4, 3.3

[20] K. Momose. Evaluation of an eye gaze point detection method using vep elicited by multi-pseudorandom stimulation for brain computer interface. In *2007 29th*

Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pages 5063–5066, 2007. doi: 10.1109/IEMBS.2007.4353478. 3.1

[21] Daiki Aminaka, Shoji Makino, and Tomasz M. Rutkowski. Eeg filtering optimization for code-modulated chromatic visual evoked potential-based brain-computer interface. In *Symbiotic*, 2015. 3.1

[22] Daiki Aminaka, Shoji Makino, and Tomasz M. Rutkowski. Svm classification study of code-modulated visual evoked potentials. *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1065–1070, 2015. 3.1

[23] Zhihua Huang, Wenming Zheng, Yingjie Wu, and Yiwen Wang. Ensemble or pool: A comprehensive study on transfer learning for c-vep bci during interpersonal interaction. *Journal of Neuroscience Methods*, 343: 108855, 2020. ISSN 0165-0270. doi: https://doi.org/10.1016/j.jneumeth.2020.108855. URL https://www.sciencedirect.com/science/article/pii/S0165027020302788. 3.1, 3.4

[24] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. Online adaptation of a c-vep brain-computer interface(bci) based on error-related potentials and unsupervised learning. *PLoS ONE*, 7, 2012. 3.1

[25] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. One class svm and canonical correlation analysis increase performance in a c-vep based brain-computer interface (bci). In *ESANN*, 2012. 3.1

[26] Martin Spüler, Wolfgang Rosenstiel, and Martin Bogdan. Unsupervised online calibration of a c-vep brain-computer interface (bci). In *ICANN*, 2013. 3.1

[27] Sebastian Nagel, Werner Dreher, Wolfgang Rosenstiel, and Martin Spüler. The effect of monitor raster latency on veps, erps and braincomputer interface performance. *Journal of Neuroscience Methods*, 295:45–50, 2018. 3.1

[28] Christoph Kapeller, Christoph Hintermüller, Mohammad Abu-Alqumsan, Robert Prueckl, Angelika Peer, and Christoph Guger. A bci using vep for continuous control of a mobile robot. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5254–5257, 2013. 3.1

[29] Faqiang Peng and Zhihua Huang. A c-vep bci system for psychological experiments. *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–5, 2019. 3.1

[30] Dewei Luo and Zhihua Huang. A subject-transfer study on detecting c-vep. *2019 Chinese Automation Congress (CAC)*, pages 2956–2959, 2019. 3.1, 3.4

[31] Hooman Nezamfar, Umut Orhan, Shalini Purwar, Kenneth E. Hild, Barry S Oken, and Deniz Erdomu. Decoding of multichannel eeg activity from the visual cortex in response to pseudorandom binary sequences of visual stimuli. *International Journal of Imaging Systems and Technology*, 21, 2011. 3.1

[32] Jordy Thielen, Philip Broek, J. Farquhar, and Peter Desain. Broad-band visually evoked potentials: Re (con)volution in brain-computer interfacing. *PLoS ONE*, 10, 07 2015. doi: 10.1371/journal.pone.0133797. 3.2, 3.5

[33] Jordy Thielen, P. Marsman, Jason D. R. Farquhar, and Peter Desain. Re(con)volution: Accurate response prediction for broad-band evoked potentials-based brain computer interfaces. In *Brain-Computer Interface Research*, 2017. 3.2, 3.5, 5.1.4, 5.3.1

[34] Ceci Verbaarschot, Daniëlle Tump, Andreea Lutu, Marzieh Borhanazad, Jordy Thielen, Philip van den Broek, Jason Farquhar, Janneke Weikamp, Joost Raaphorst, Jan T. Groothuis, and Peter Desain. A visual brain-computer interface as communication aid for patients with amyotrophic lateral sclerosis. *Clinical Neurophysiology*, 132(10):2404–2415, 2021. ISSN 1388-2457. doi: https://doi.org/10.1016/j.clinph.2021.07.012. URL https://www.sciencedirect.com/science/article/pii/S1388245721006635. 3.2

[35] Sebastian Nagel, Martin Spüler, and Wolfgang Rosenstiel. Random visual evoked potentials (rvep) for brain-computer interface (bci) control. 2017. 3.3

[36] Sebastian Nagel, Martin Spüler, et al. Asynchronous non-invasive high-speed bci speller with robust non-control state detection. *bioRxiv*, 2018. 3.3, 3.5

[37] Qingguo Wei, Yonghui Liu, Xiaorong Gao, Yijun Wang, Chen Yang, Zongwu Lu, and Huayuan Gong. A novel c-vep bci paradigm for increasing the number of stimulus targets based on grouping modulation with different codes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(6):1178–1187, 2018. doi: 10.1109/TNSRE.2018.2837501. 3.3

[38] Eduardo Santamaría-Vázquez, Víctor Martínez-Cagigal, Sergio Pérez-Velasco, Diego Marcos Martínez, and Roberto Hornero. Robust asynchronous control of erp-based brain-computer interfaces using deep learning. *Computer Methods and Programs in Biomedicine*, 215:106623, 01 2022. doi: 10.1016/j.cmpb.2022.106623. 3.3, 3.4, 3.6, 5.3.1

[39] Jiahui Ying, Qingguo Wei, and Xichen Zhou. Riemannian geometry-based transfer learning for reducing training time in c-vep bcis. *Scientific Reports*, 12:9818, 06 2022. doi: 10.1038/s41598-022-14026-y. 3.4

[40] Jun-ichi Sato and Yoshikazu Washizawa. Reliability-based automatic repeat request for short code modulation visual evoked potentials in brain computer interfaces. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 562–565, 2015. doi: 10.1109/EMBC.2015.7318424. 3.5

[41] Eduardo Santamaría-Vázquez, Víctor Martínez-Cagigal, Javier Gomez-Pilar, and Roberto Hornero. Asynchronous control of erp-based bci spellers using steady-state visual evoked potentials elicited by peripheral stimuli. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(9):1883–1892, 2019. doi: 10.1109/TNSRE.2019.2934645. 3.6

[42] Robert Gold. Optimal binary sequences for spread spectrum multiplexing (corresp.). *IEEE Trans. Inf. Theory*, 13:619–621, 1967. 4.1, 4.2

[43] S Ahmadi, M Borhanazad, D Tump, J Farquhar, and P Desain. Low channel count montages using sensor tying for VEP-based BCI. *Journal of Neural Engineering*, 16(6):066038, nov 2019. doi: 10.1088/1741-2552/ab4057. URL `https://doi.org/10.1088/1741-2552/ab4057`. 4.2

[44] Siebe Geurts. A deep learning approach to noise tagging. 2021. 5.1.2, 5.1.3

[45] Julia Janssen. Exploring code families and event-types for cvep bcis. 2021. 5.1.2, 5.1.3

[46] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`. 5.2

[47] Eduardo Santamaría-Vázquez, Víctor Martínez-Cagigal, Fernando Vaquerizo-Villar, and Roberto Hornero. Eeg-inception: A novel deep convolutional neural network for assistive erp-based brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28:2773–2782, 2020. 5.3.1

[48] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019. doi: 10.1007/s11263-019-01228-7. URL `https://doi.org/10.1007%2Fs11263-019-01228-7`. 5.3.3

[49] Aylin Kolba and Aydn Ünsal. A comparison of the outlier detecting methods: An application on turkish foreign trade data. *Journal of Mathematical Sciences*, 5: 213–234, 08 2021. 5.3.5