Tackling Drift in Neural Responses in the Spinomotor Pathway

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Hrithik Nambiar
ID No. 2019A7PS0100G

Under the supervision of:

Dr. Thomas Serre
&
Dr. Neena Goveas

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS
September 2023

Declaration of Authorship

I, Hrithik Nambiar, declare that this Undergraduate Thesis titled, 'Tackling Drift in Neural Responses in the Spinomotor Pathway' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____________________________________

Date: ____________________________________

Certificate

This is to certify that the thesis entitled, "*Tackling Drift in Neural Responses in the Spinomotor Pathway*" and submitted by Hrithik Nambiar ID No. 2019A7PS0100G in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor
Dr. Thomas Serre
Professor,
Brown University and ANR-3IA Artificial and Natural Intelligence Toulouse Institute
Date:

Co-Supervisor
Dr. Neena Goveas
Professor,
BITS-Pilani Goa Campus
Date:

"The measure of intelligence is the ability to change."

Albert Einstein

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, GOA CAMPUS

Abstract

Bachelor of Engineering (Hons.)

Tackling Drift in Neural Responses in the Spinomotor Pathway

by Hrithik NAMBIAR

This thesis examines the problem of neural drift and the possibility of using meta-learning to build an adaptive algorithm to tackle this drift. Spinal Cord Injury(SCI) is a debilitating condition that can result in a wide range of physical and neurological impairments. Recent research has identified Epidural Electrical Stimulation(EES) as a promising approach for restoring motion in SCI patients. This is preliminary work as a part of our goal to build an efficient and adaptive neural network-based solution for identifying EES parameters which initiate the desired motion in patients with SCI. Reliability and Efficiency are key to adopting such a solution for clinical purposes. Hence, the neural network should be able to adapt to neural drift to maintain its performance using minimal trials for re-calibration. The method followed includes data collected using experiments spanning six days in which a sheep was surgically implanted with a multi-electrode array through which electrical stimulations were applied. The activity of the sheep's hind legs was collected using surface electromyography (EMG) sensors. We compare the performance of reliable baselines and our adaptive algorithms on a held-out set from the sixth day of the experiments. The goal of the adaptive algorithm is to minimize the trials used from the sixth day for adaptation while maintaining the prediction performance. The results indicate that pre-training on the data collected from the first five days of the experiments can aid performance to perform at par with the baseline by using only half the number of trials. Although the results indicate that rudimentary meta-learning algorithms such as Model-Agnostic Meta-Learning(MAML) and First-Order Model-Agnostic Meta-Learning(FOMAML) fail to outperform baselines, we find more recent meta-learning algorithms promising for alleviating the challenges identified in this thesis.

Acknowledgements

I would like to thank my thesis supervisors, Dr Thomas Serre and Dr Neena Goveas, for their invaluable guidance and constant support throughout this project. I would especially like to thank Dr Jaeson Jang (Post-Doctoral research associate, Serre Lab at Brown University) for his guidance, support, and mentorship and for helping me track my progress in the project. Our weekly meetings and discussions were instrumental to the completion of this thesis and I am looking forward to our future work on this project in the lab. I am also grateful to Mathieu Chalvidal (PhD student, ANITI & Serre Lab) for his discussions relating to my project. I would also like to thank David Borton, Jonathan Calvert, and Samuel Parker from the Borton Lab (at Brown University) for their contributions to data collection and facilitating access to the datasets used in this study.

To all my friends that I have made throughout my university career, thank you for being my family away from home. Finally, my eternal gratitude to my family for always being there for me and for their constant support and encouragement.

Contents

List of Figures

List of Tables

Abbreviations

EMG	**E**lectro**m**yo**g**raphy
EES	**E**pidural **E**lectrical **S**timulation
SCI	**S**pinal **C**ord **I**njury
BCI	**B**rain-**C**omputer **I**nterface
MAML	**M**odel-**A**gnostic **M**eta-**L**earning
FOMAML	**F**irst-**O**rder **M**odel-**A**gnostic **M**eta-**L**earning

Dedicated to the reader.

Chapter 1

Introduction

Spinal Cord Injury (SCI) is an impairing condition that can result in a wide range of physical and neurological impairments. One reliable approach for restoring motor function in patients with SCI is using Epidural Electrical Stimulation (EES). EES involves applying electrical stimulations to the spinal cord using an electrode array implanted in the spinal cord. Research has proven that these electrical stimulations help activate motor neurons and muscles, allowing for several voluntary movements including complex activities such as cycling or swimming for a long period after SCI. Recent research[14] has shown that it is possible to leverage modern deep neural networks to optimize the large neurostimulation parameter space for predicting motor outputs. This demonstrates the ability of these networks to approximate spinal sensorimotor computations. While this study has shown promising results, this thesis identifies and attempts to remediate a critical challenge in implementing EES as a solution for SCI patients: the failure of machine learning models due to neural drift.

Although deep learning models are good approximators of spinal sensorimotor computations, they are susceptible to changes in neural activity over time, leading to their declining performance. We refer to this change over time in neural activity as neural drift. Various reasons can be attributed to the observed drift in neural activity collected from the implanted electrodes such as tissue growth around the electrode array implants or external factors such as electrode position and environment. Neural drift poses a significant obstacle to the practical implementation of a neural network-based EES for SCI patients. It can require frequent re-calibration and retraining of the deep learning models, reducing their usability and practicality for clinical purposes.

To address the problem of neural drift in deep neural networks for EES, it is imperative to develop adaptive algorithms to help the neural network adapt to the changing neural activity efficiently. One such promising direction for adaptation to neural drift is Meta-Learning, which refers to the idea of training a network to learn how to learn. In this training regime, the model learns from a distribution of tasks, rather than a specific task, and uses this knowledge to generalise to a new task efficiently. Meta-learning has shown tremendous success and applicability in domains such as Images, Language and Robotics. We introduce meta-learning to the realm of neural engineering as an efficient solution for the neural drift observed in biological neural data.

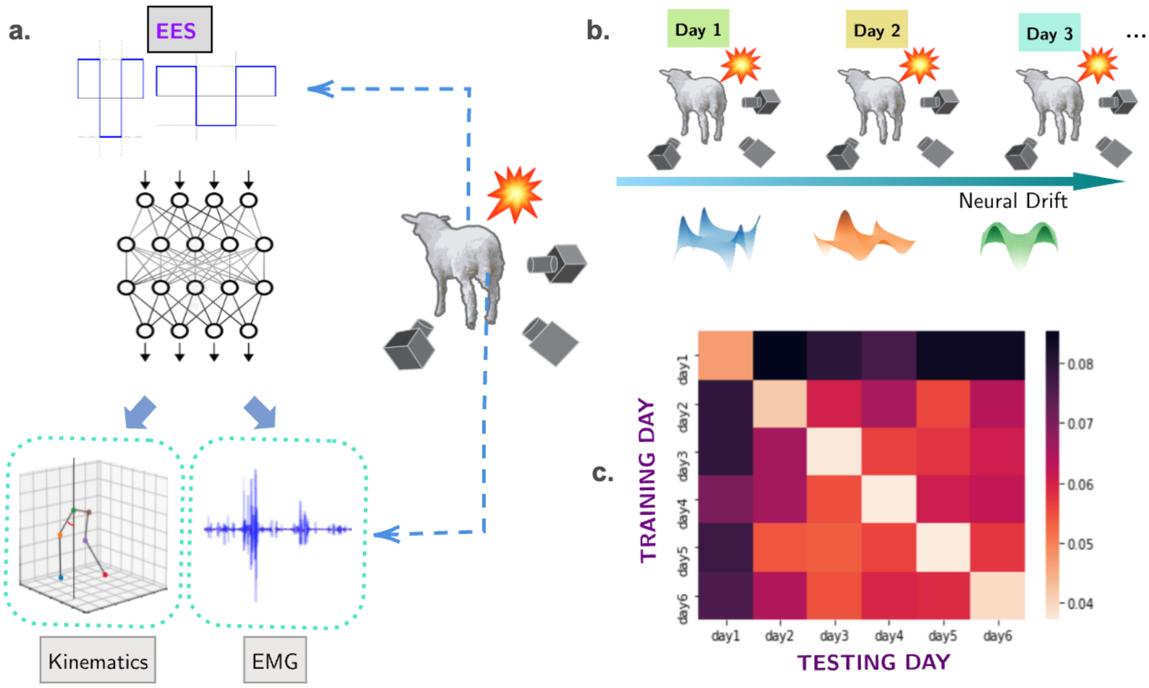

FIGURE 1.1: Goal: To build an adaptive learning algorithm to help tackle neural drift and maintain model performance by using minimal training trials. (a) EMG and kinematics responses to varying EES applied to the sheep's spine are collected. A neural network can be trained to predict them using the EES parameters. (b) Data is collected over 6 consecutive days. (c) The forward model which predicts the power of EMG responses fails to maintain its performance when tested on a day different from when it was trained on.

1.1 Project Goals

In this work, we analyse the data collected from a sheep surgically implanted with two 24-contact EES electrode arrays placed onto the spinal cord[3]. Experiments involved different stimulation conditions repeated multiple times and the resulting muscle activity was measured via surface electromyography(EMG) from four bilateral lower extremity muscles collected over 6 consecutive days. This data is used to analyse the presence of drift in the surface EMG that was recorded post-neural stimulation and further test adaptive algorithms as a solution. Further, the synchronized video recording of the hind leg of the sheep was obtained using three fixed cameras. This data was utilised along with state-of-the-art Computer Vision algorithms to extract the Kinematics of the sheep as a response to the stimulation. This thesis also recognises the drift observed in kinematics as a response across days with similar experimental conditions and attempts to remediate the same using meta-learning algorithms.

1.2 Structure of the report

Chapter 1 is the introductory chapter, where the motivation behind the project and the project goals and contributions are presented. Chapter 2 explores and discusses previous literature available on the concept drift and neural drift, solutions to mitigating the drift seen in neural data and state-of-the-art meta-learning algorithms to motivate the readers on their potential usability for this project. Chapter 3 introduces the dataset collected for this project, the analysis of the dataset, the meta-learning approach suggested and the other baselines. Chapter 4 presents, the results of the analysis and the method will then be evaluated through a series of experiments with other baselines. The obtained results are compared and discussed in this chapter. Finally, the last chapter provides a general summary outlining the project's key takeaways and identifies improvement points for immediate future work.

Chapter 2

Related Works

2.1 Concept Drift

Concept drift and data drift are two major challenges in machine learning, especially while deploying trained models for real-world applications. Concept drift refers to the changes in the relationship between the input and target variables over time and data drift refers to the changes in the input distribution. These changes can lead to degradation in the performance of machine learning models since it now encounters data that is different from the data that it was trained on. This makes them less accurate and less reliable. Recent years have seen a growing interest in developing methods to detect concept and data drift especially for deployed machine learning models where it is important to maintain their performance[21]. The literature contains various methods to detect drift by calculating the distance between the training and test distributions, prediction errors on the new test data and the changes in the accuracy of the deployed model. However, the most critical challenge is to adapt to these changes post-detection. Saadallah and Morik[28] proposed a meta-learning-based approach to learn the weights of an ensemble model to adapt to concept drift. Diez-Olivan et al[8] propose the use of kernel density estimation techniques for generating synthetic data for finetuning the model to adapt to concept drift. DAREM[6] proposed an algorithm based on incremental learning to update the model parameters for handling the drift. It alleviates the problem associated with catastrophic forgetting by including both the old and new data. To avoid retaining old data, Fekri et al[11] introduce a tree-structured Parzen estimator or optimising hyperparameters and also the weights of a recurrent neural network for concept drift adaptation.

2.2 Neural Drift

Research has revealed that neural activation associated with motor functions, sensation and cognition undergoes changes over days and weeks, which we refer to as neural drift or representational drift. Studies have revealed continual reorganisation of neuronal activity pertaining to certain tasks even when the task has been fully learnt[26]. Driscoll et al[10] designed a sensorimotor task involving a mouse which was trained to navigate a maze set in a virtual environment. It was found that the neural activity in the posterior parietal cortex (PPC) of the mice which was required to solve the task, was not stable across days and weeks. Similarly drift in neural activity has been observed in the motor cortex[15] during sensorimotor tasks.

Representational drift was previously seen as a passive process caused by noise, however, it can also be influenced by ongoing learning. A small fraction of neurons exhibits discrete changes in activation from one day to the next or over a shorter time interval. Hence, the correlation of neural activity with observed behavioural and sensorimotor variables usually remains stable over shorter intervals. However, over the course of a day or weeks, these changes accumulate to a higher level such that the changes in correlation between the neural responses and the observed variables change[9]. In typical experiments, the neural activity associated with a specific task is analysed, but the same population of neurons could be associated with other aspects, such as the subject learning new experiences through the course of the experiment. Even in the absence of learning, studies have shown that continual reorganisation in various parts of the brain for encoding information more efficiently.

A part of the literature[22] also presents theoretical works that highlight the computational benefits of representational drift and also how it supports robust actions, perceptions and memories. One challenge that could arise due to the changing activity could be the decoding ability of the nervous system. However, theoretical work[27] has also shown that drift can be compensated by synaptic plasticity which requires only small weight changes over days enabling animals to maintain a stable readout of information by a downstream network. Similarly, it is vital to implement adaptive mechanisms in artificial neural networks that have been widely deployed for decoding neural activity for various tasks to handle drift making them robust and reliable.

2.3 Adaptive Algorithms for neural decoders

Neural drift has been a big challenge in implementing closed-loop solutions for neural decoding for a decade. There have been interesting Adaptive learning-based neural decoding systems that have been introduced in the literature. Dantas et al[5] presented a semi-supervised adaptive learning algorithm for movement intent decoding using electromyography (EMG) signals. They utilise a movement model which recognizes the recently decoded movement patterns and use this information to retrain the decoder in a semi-supervised manner. Tafazoli et al[32] recognised the need for an adaptive-closed-loop stimulation system (ACLS) to control the drifting pattern activity of a population of neurons. ACLS uses a model-free learning algorithm which involves a stochastic learning algorithm that helps reduce the error between the evoked and target neural responses. Chiang et al [4] suggest using error-related potentials to maintain the performance of Brain-Computer-Interface frameworks using Electroencephalography (EEG), to make them robust to drifts. Error potentials are calculated shortly after the output of the classifier is displayed to the user by using the EEG responses of the user in this closed-loop framework. There are multiple works[16][7] which suggest using the lo dimensional latent manifold associated with neural population for stabilising BCI. Degenhart et al. introduced a stabilized BCI which utilizes the low-level structure present in neural activity to enable accurate control even in the presence of drift or instabilities. This work is based on the recent findings that point to the evidence that neural population activity tends to lie in a low-dimensional space termed the neural manifold. This work introduces a manifold-based stabilizer based on factor analysis whose parameters are updated across time to account for the instabilities and a fixed decoder. A recent work by Bonizatto et al[2] presents a promising approach using Gaussian-Process (GP) based Bayesian-Optimisation (BO) which can control stimulation parameters for motor cortex and spinal cord outputs. They show that it can also deal with challenges associated with drift by optimizing stimulation using limited data and purely online. There have not been enough works that suggest the application of meta-learning for tackling neural drift. A work by Li et al[19] presents a meta-learning-based approach for EEG motor imagery decoding to assist a BCI decoder to generalise between users and recording sessions.

2.4 Meta-Learning

Meta-Learning is a learning paradigm in machine learning which aims to improve the generalisation of models such that they are capable of adapting to new tasks and environments that are unseen during the training phase. It is often associated with the idea of "learning to learn". In this paradigm, the network is trained over a distribution of tasks and its performance is measured on a new unseen task. Over the years, there has been a lot of interesting literature in this area. We can broadly classify these works into three categories - metric-based, model-based and optimisation-based meta-learning.

2.4.1 Metric-Based

The main idea behind this type of learning method is to learn a metric space where the embedding of samples from similar tasks or environments is closer to each other. The idea is similar to nearest neighbour algorithms. A metric function is learnt to compare these tasks. Once this has been learned, it can be used to quickly adapt to new tasks by finding tasks similar to the new task. Koch et al[18] proposed using Siamese neural networks for one-shot image classification. Siamese neural networks consist of two neural networks which are jointly trained using a function so that it learns if two images belong to the same class. During the test time, an image is assigned the label associated with the image from the support set that is closest to it in the metric space. Some other notable works are matching[33], prototypical[30], and relation neural networks[31].

2.4.2 Model-Based

This section reviews works that present models designed for learning new tasks efficiently and rapidly. Memory Augmented Neural Networks (MANN)[29] introduced by Santoro et al uses an external memory which can be accessed and modified during both training and testing time to quickly adapt to new tasks. Munkhdalai et al proposed MetaNets[24] which consists of two learners, namely the meta and base learners. The base learner attempts to solve specific tasks and the meta learner operates across tasks and rapidly changes the model weights. Conditional Neural Processes(CNP)[13] was proposed to combine the advantages of deep neural networks and Gaussian processes for allowing fast inference by taking advantage of prior knowledge.

2.4.3 Optimisation-Based

This section reviews optimisation-based meta-learning which treats meta-learning as an optimisation problem, which can then be tackled using gradient descent. Model Agnostic Meta-Learning[12] aims to learn model parameters such that it can adapt to new unseen tasks using only a few of steps gradient descent and using only a few examples. Multiple works were built on top of MAML for improving its performance. Nichol et al[25] suggested improving this algorithm by omitting the second derivatives involved in the MAML. They also introduced a new algorithm called Reptile which involves repeatedly sampling a task and moving the initialisation towards the trained weights for that task. Alpha MAML[1] introduced an adaptive algorithm to adjust the hyperparameters involved in the original algorithm and thereby improving its performance. Li et al[20] proposed Meta SGD which is a meta-learner which can initialise and adapt to any differential learner in just one step.

Chapter 3

Methodology and Experiments

3.1 Dataset

This dataset has been collected through experiments spanning 6 days (Monday to Friday from week 1 and Monday from week 2) and to test the presence of inter-day drift and to facilitate an attempt to build adaptive algorithms for building a reliable BMI. A fully conscious sheep surgically implanted with EES electrode arrays was hanged in the air using a sling until a separation between the hooves and the ground was observed. The electrodes were implanted in the dorsal aspect of the epidural space of the lumbosacral spinal cord, spanning approximately the L4–L6 vertebral bodies[3]. The sheep was allowed recovery time before conducting the experiments.

3.1.1 Data Acquisition

Muscle activation and the kinematics (trajectories of hip, knee and toes of the hindlegs of the sheep) as a response to the electrical stimulation were collected. Muscle activation was collected using the surface EMG of the target muscles. The target muscles included lower extremity muscles on both the hindlegs of the sheep: peroneus longus (PL), biceps femoris (BF), gracilis (GR) and gastrocnemius (GA). The target muscles are selected after considering the positive experimental results previously obtained by the team. BF is a proximal muscle responsible for hock flexion and hip extension, and GR is a proximal muscle responsible for hip abduction, hock flexion, and hock internal rotation. PL is a distal muscle responsible for hock flexion and GA is a distal muscle responsible for hock extension.[14] The EMG sensors were placed after careful preparation of the attachment sites.

Randomised electrical stimulations consisting of 5 unique amplitudes (300μA, 600μA, 900μA, 1200μA, 1500μA) and 4 unique frequencies (10Hz, 25Hz, 50Hz, 100Hz) were delivered through

36 unique electrodes implanted in the spinal cord. On day 6 the experiment involved 5 unique amplitudes with the maximum being 1800μA. Electrical stimulation pulses of 300ms were delivered every 1 second of the experiment. The EMG signals collected were bandpass filtered and digitized during data collection. The EMG signals from 500ms before stimulation to 500ms were collected for each trial involving a specific EES parameter (Amplitude, Frequency and Electrode). Experiments involving a specific EES parameter set were repeated 3-5 times.

Additionally, the synchronised video recording of the hind legs of the sheep was also collected during the experiment for analysing the kinematics of the motion elicited by the stimulations. The video was collected using 3 fixed cameras to make the recordings consistent across the 6 days of the experiment. The kinematics of the hindlegs of the sheep was extracted from the video using state-of-art computer vision techniques as highlighted in section 3.1.2.

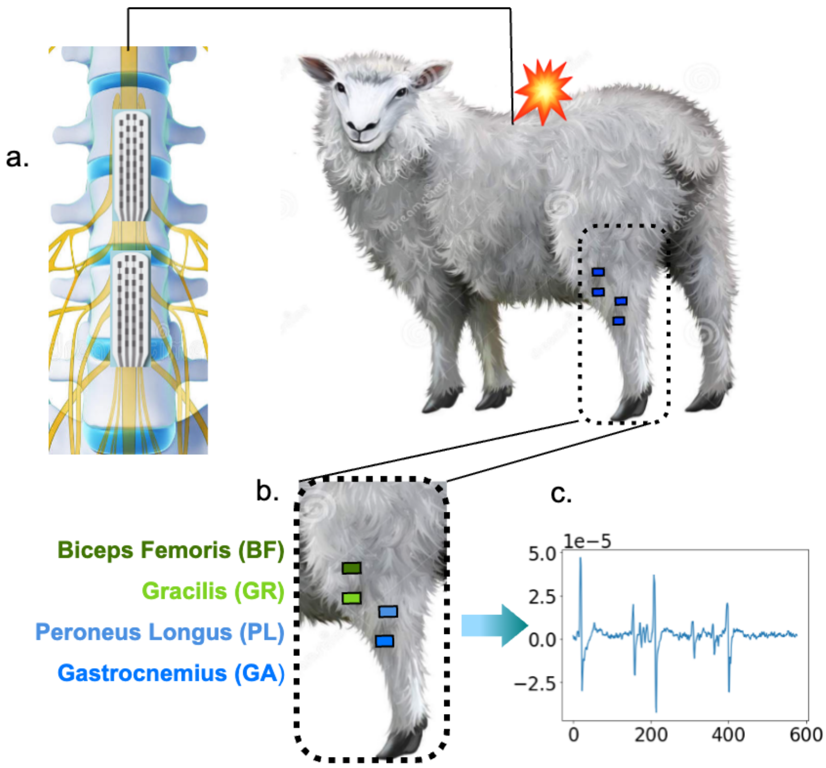

FIGURE 3.1: (a) Multi-array electrode array implanted in the spinal cord of a sheep. (b) EMG sensors are attached to 4 muscles of each hindleg. (c) The surface EMG of these muscles in response to EES is collected.

3.1.2 Data Processing

EES for each trial consists of three parameters, the amplitude (A), the frequency (F) and the electrode (E). For using this as the input for neural networks, the EES was summarised as $\theta = [A, F, E]$; where E is the one-hot encoding of the electrode number e [that is, with number 1 at the index e and 0 elsewhere]. This makes θ a vector of dimension 38 (since we have 36 electrodes used to create the dataset).

The EMG waveforms were collected from -500ms prior to the stimulation till 500ms after the stimulation. EMG was filtered using a high pass filter to 100Hz to prevent any noise caused by the movement of the artefacts used in the experiment. The EMG responses collected are pre-processed by removing outliers and unreliable samples as described by previous research by Govindarajan et al[14]. For summarization of the EMG responses to a lower dimension for the ease of evaluating a neural network-based solution, we adhere to the guidelines set by prior research works. We consider the average power of the EMG response elicited in the muscle from 100ms to 400ms post-stimulation. The average power is obtained by averaging the absolute values of associated EMG responses in this interval. We chose this interval since it is a true representation of the response elicited by the stimulation.

Using the video recordings collected from the three cameras in the experiments, the kinematics of the thigh, knee and toe from the sheep's hind legs were extracted using a state-of-the-art pose estimation library called DeepLabCut[23]. Further processing was performed to extract annotations with accuracy at par with human annotators.

3.2 EMG Analysis

EMG responses to electrical stimulations were analysed to check if there was an inter-day drift. To determine this, the inter-day and intra-day changes in the average power of the EMG responses as well as the raw EMG responses in different muscles to different EES were compared. One of the major goals of this analysis was to determine if certain kinds of electrical stimulation resulted in more inter-day drift when the same experiment was conducted across days.

3.2.1 Drift in the EMG responses

It was observed that there was an inter-day drift in the power of the EMG responses from 100-400ms post the stimulations, given the same stimulation conditions. This was considerably more than the changes observed during the multiple trials for the same stimulation parameters conducted on a single day (inter-day drift). This observation motivates us to ponder if a forward model that predicts the average power of EMG response trained on a single day would yield results at par with the day that it was trained on when tested on other days.

To better visualize the magnitude of the drift in the response after stimulation, it is often better to project the data to a lower dimension. The principal component analysis (PCA) is a great tool for this. We plot the data projected along the first two principal components. For similar

EES parameters, clusters of trials from each day separated from each other were observed. This verifies the presence of inter-day drift.

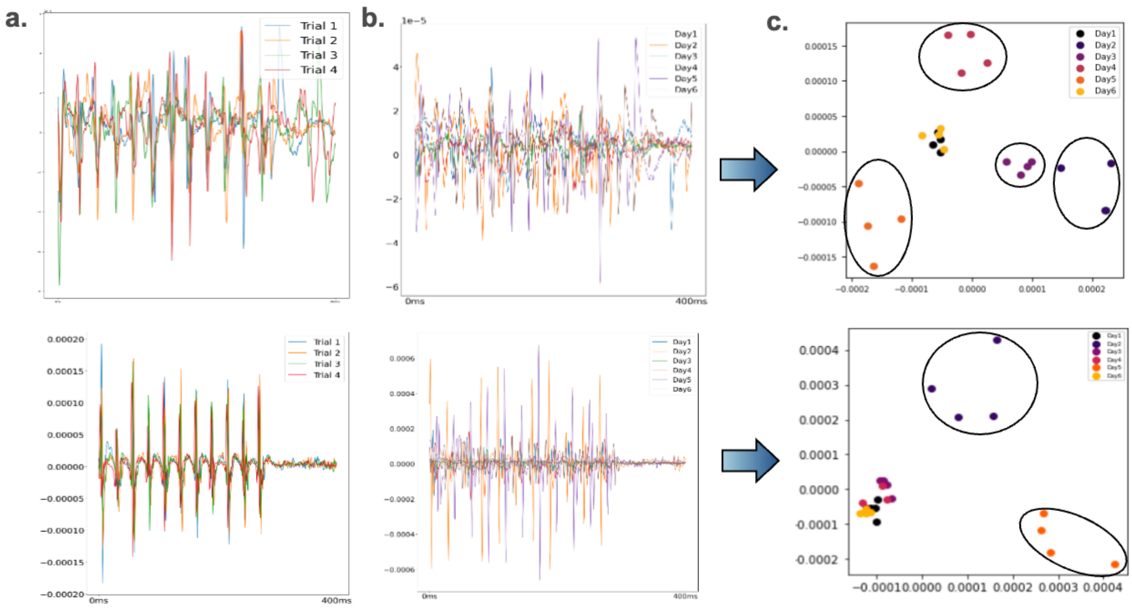

FIGURE 3.2: (a)The EMG responses from 0-400ms post EES are similar in the 4 trials involving the same EES parameters. (b)EMG responses are different for the same EES parameters when applied on different days (inter-day drift). We refer to this as neural drift. (c)Drift is more evident when visualised in a lower dimension using Principle Component Analysis(PCA). Clusters involving trials from the same day are away from clusters from other days.

3.2.2 Forward model

In this work, a multi-layer perceptron was used as the forward model for the prediction of EMG responses using EES parameters. This model was trained end-to-end using stochastic gradient descent employed in the Adam optimizer[17]. The network is implemented as three modules, the embedding, the core and the readout module. The network takes as its input the parameterized EES θ = [A, F, E]; where E is the one-hot encoding of the electrode number e [that is, with number 1 at the index e and 0 elsewhere]. This makes θ a vector of dimension 38 (since we have 36 electrodes used to create the dataset). This input is then projected onto a feature space which is then fed into the core module. The readout module uses these processed features to predict the summarized EMG responses (power of responses). The dimensions of the embedding and the core modules were 32 and 256 respectively, and the readout layer was of 8 dimensions for predicting the EMG responses in 8 target muscles (4 in each hindleg). The network applies ReLU activation function to the output of each layer. This architecture was inspired by the architecture used in the earlier research work[14]; hence, no further tuning of hyperparameters involved in the architecture was performed. We also use a Dropout layer with 50% dropout probability before the readout module to avoid overfitting and for regularization. The network is trained by minimising the L1 loss (using the mean of the L1 loss across target muscles) in the predicted and the true EMG response summaries.

The optimization algorithm used to train the model was the Adam optimizer[17] with a learning rate of 0.001 and with weight decay of 5×10^{-4} for regularisation.

The train-test split utilised is similar to the approach proposed by Govindarajan et al.[14], refer to 3.2.4 for more details. For all the results we used a holdout set consisting of 900μA amplitude and 50Hz frequency. Hence 40% of the available trials from each day are used to evaluate the model trained on the remaining 60% trials on the day. This particular amplitude and frequency were selected for the held-out set because this stimulation is not strong enough to saturate the responses and not too weak to elicit weak EMG responses.

3.2.3 Failure of forward model

The analysis of the EMG responses revealed an inter-day drift in the responses for similar stimulation conditions. It is important to study if this drift affects the performance of a model trained on a particular day when tested on other days. We train the forward on each day with the same held-out set (900μA, 50Hz) and compare its performance on the held-out sets (900μA, 50Hz) from the other 5 days of the experiment. We observed the L1 error on the held-out set is considerably higher (almost 2 times the error on a training day) when tested on the other days. This is the motivation behind developing adaptive algorithms as discussed in the remaining part of this report.

3.2.4 Train-Test Split

We use a similar train-test split as introduced in previous work[14]. This scheme has been crafted to test the generalisation ability of our forward model as well as our adaptive algorithms. We want to test both (a)hard generalisation: The model's ability to interpolate to completely new parameters and (b) soft generalisation: the model's ability to interpolate along one unseen parameter. For building a forward model specific to a day, we can utilise all the trials from that day except the test set from the day. This is our baseline. The aim of this project is to build a adaptive algorithm, which can perform at par or even outperform this baseline forward model by using a smaller number of trials from the same day.
For building and testing our adaptive algorithms, we utilise all trials from day 1 to day 5 (which includes experiments from the same week) to train the adaptive algorithm. We use two fine-tuning or adaptation strategies involving data from day 6 (the test day in general) which we name as Adaptation 1 and 2 respectively. Adapt 1 involves 252 trials, which are sampled from

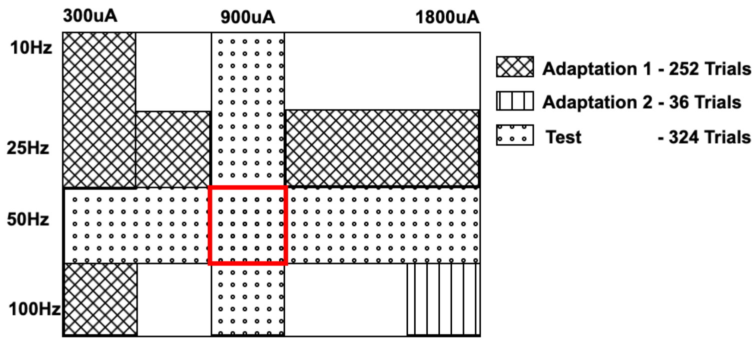

Figure 3.3: The test set and the adaptation sets which were used to test the fine-tuning ability of adaptive algorithms. The red box highlights the part of the test set used to test the hard-generalisation ability of the model.

any one frequency or one amplitude as represented in the figure. Adaptation 2 involves only 36 trials, which are sampled from a unique frequency and a unique amplitude. We would like to develop algorithms which can perform as well as the model trained specifically for day 6 by using a trained adaptive model which is fine-tuned to day 6 using either of these adaptation sets designed.

3.3 Meta-Learning

A forward model trained to predict EMG responses using experimental data from a day fails to maintain the same level of performance when tested on data from the following days due to the presence of neural drift. This implies that to maintain the same level of performance as the training day on any other day, the model would require to be retrained using data collected from that particular day. However, it is difficult and not practical to conduct enough trials every day for retraining. For this reason, we would like to implement an adaptive learning mechanism which would help to reconfigure the parameters of the forward model using only a few trials from each day to maintain a consistent model performance across days. In this work, we explore a meta-learning-based approach for adapting to drift.

Meta-learning is a training paradigm that is different from contemporary machine learning algorithms which are trained from scratch to optimise the performance on a specific task. Meta-Learning targets ‘learning-to-learn’ or in other words attempting to learn the learning algorithm itself. It does this by training itself over multiple learning episodes using a distribution of related tasks and uses this to improve its future learning processes. In the context of our research problem, we view the data collected from the first five days as different tasks. The assumption is that these “tasks” are related and thereby the optimal parameters would be closer to each other.

Over multiple learning episodes, it is expected that the model learns to converge to a point in the parameter space, from where it is easy to adapt to the drifted data from a new day using only a smaller number of samples or lesser gradient descents.

3.3.1 Model-Agnostic Meta-Learning

Just like other commonly used meta-learning algorithms, the idea behind MAML[12] is to directly optimize for the initial representation such that it can be effectively fine-tuned using only a few labelled samples. MAML is trained on a wide variety of tasks (data collected from different days in the context of this project) so that it can learn the best representation. This representation is trained such that it can quickly adapt to a new task via a few gradient descent steps efficiently in a few-shot fashion. The intuition behind using this approach in this project for tackling the challenge of neural drift is that there must be internal features of the data that could be transferred across different tasks. In effect, through the training procedure, we aim to find model parameters such that small changes in these parameters would yield large improvements in the loss function on different tasks when steps are taken along the gradient of that loss. Since this is a gradient-based approach, we assume that the loss function is smooth.

Algorithm MAML Algorithm for tackling neural drift

Require: $p(\mathcal{T})$: Set of data from all training days
Require: Iterations, Adaptation Steps, Inner learning rate(α), Meta-learning rate(β)
for `Iterations` **do**
    Set of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
    for each $\mathcal{T}_i$ **do**
        Sample Train-Set $Train_i$ and Held-out set $Test_i$ from $\mathcal{T}_i$
        for `Adaptation steps` **do**
            Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to $Train_i$
            Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
            Compute the L1 evaluation loss $Eval_i$ using $Test_i$
        end for
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} Eval_{\mathcal{T}_i}$
end for

For applying MAML to this project, we view the data collected from each day as the different tasks belonging to the same task distribution T. The training task set consists of data from the first 5 days of the experimental data. Our aim by using meta-learning is to learn the best initialization θ, such that it becomes easy for fine-tuning. We report the results on the held-out set as mentioned in 3.2.4, that is, (900μA, 50Hz) from Day 6. This is done so that we can compare its performance with the forward model introduced in section 3.2.2. We compute the task-specific parameters θ' for data from each day using gradient descent for multiple epochs (adaptation steps) with a learning rate α. After the task-specific model has been trained, we compute the loss of this model on the held-out set from that specific day to yield $Eval_i$. This loss is aggregated and averaged across all tasks (5 days) for performing the meta update.

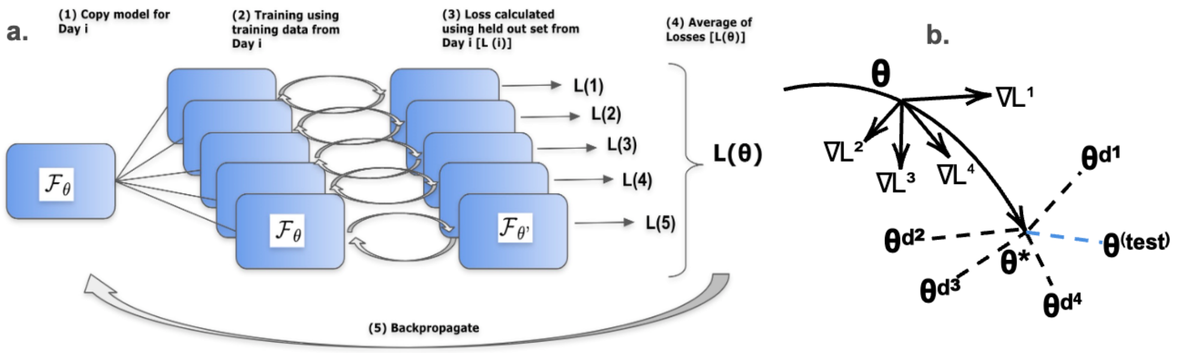

FIGURE 3.4: (a) A pictorial representation of the MAML algorithm to tackle neural drift. (b) The algorithm optimizes for the model parameter θ. Through meta-update the initialisation θ* which is good for fine-tuning is learnt.

The meta-update is where the parameters of the model are changed. We perform meta-optimisation using the Adam Optimizer[17] with a different learning rate β. This is performed for multiple iterations, which we refer to as learning episodes. The loss used is the mean L1 error.

3.3.2 First-Order Model-Agnostic Meta-Learning

The meta-optimization involved in MAML involves a second-order derivative for performing meta-update. In contemporary applications, MAML is used in a few-shot setting where only a few examples are available for the tasks involved in the training phase. However, we have sufficient samples available for days from 1 to 5, which we would like to utilize. Hence, we use a large number of adaptation steps so that we learn very good task-specific learners. This combined with the need for calculating the second derivative, makes the MAML algorithm very slow for the meta-learning episodes to be increased in the quest for better results. First-order MAML (FOMAML)[25] is a cheaper and simplified implementation of MAML which omits the second derivatives.

3.4 Baselines

We compare the performance of MAML on the held-out dataset (900μA,50Hz) from day 6 with two baselines. The first is a forward model trained using all the training data from day 6 (remaining trials except the trials in the held-out set). The second is a forward model trained on all the trials collected from the experiments from day 1 to day 5. This can be viewed as analogous to transfer learning or using pre-trained weights. The goal of this work is to build an adaptive learning algorithm that can outperform or even be at par with the baselines using lesser trials from day 6. Hence, we compare the performance of the pre-trained and MAML-based models on their prediction ability on the held-out set after adaptation or fine-tuning using the two adaptation strategies highlighted in figure 3.3.

Chapter 4

Results

4.1 Results

Results reported are the average L1 error (for the 8 muscles on both hindlegs of the sheep) for the prediction of the power of EMG responses on a held-out set consisting of 324 trials (900μA,50Hz) from Day 6. Days 1 to 5 in the dataset are from Monday to Friday, and Day 6 was Monday from the week after. For this reason, data from day 6 was used for testing the adaptive algorithms (refer Table 4.1 and Table 4.2) since it is expected to have the maximum drift. However, we make sure to perform cross-validation of these results by testing on each of the 6 days whilst training on the remaining 5 days (Figure 4.1). We choose 900μA and 50Hz for the held-out set since this stimulation does not belong to either extremity and hence is good to test the interpolation and generalisation ability of the prediction model. The baselines for our meta-learning algorithm are described in section 3.4. As introduced in section 3.2.4 our aim is adaptation using a smaller number of trials and hence we test the performance by fine-tuning using two types of finetuning datasets, namely, Adaptation 1 and 2.

4.1.1 Pre-training

This section will focus on the results obtained by pre-training on the trials conducted from Day 1 to Day 5. The L1 errors for an MLP trained on day 6 and the pre-trained model are shown in Table 4.1. As expected, pre-training helps the model learn a suitable initialisation of the model parameters. The pre-trained model is able to leverage some features from the data from the earlier days to help it perform slightly better than a forward model trained only on trials from day 6 (for 1000 epochs) by using the same number of trials (540 trials) to finetune its weights (for 100 epochs). A pre-trained model is also able to perform almost at par (L1 error of 0.0423 compared to 0.0412) by using only half the number of trials (252 trials - Adaptation 1) from

Model	**Training Data**	**Finetuning Data**	**L1 Error (On held-out set from Day 6)**
MLP	540 Trials from Day 6	-	0.0412
Pre-Trained MLP	All Trials from Day 1 to Day 5	-	0.0512
		540 Trials from Day 6	**0.0385**
		252 Trials (Adaptation 1) from Day 6	0.0423
MAML	All Trials from Day 1 to Day 5	252 Trials (Adaptation 1) from Day 6	0.0469
First-Order MAML	All Trials from Day 1 to Day 5	252 Trials (Adaptation 1) from Day 6	0.0493

TABLE 4.1: Comparison of the performance of the baselines and the meta-learning algorithms (after fine-tuning using Adaptation 1) set on the held-out set from Day 6.

day 6 for fine-tuning. However, it is not as accurate as the forward model for day 6 without fine-tuning using trials from day 6. But it is to be noted that a pre-trained model performs reasonably well on the held-out set from day 6 without using any data from the day.

4.1.2 Adaptation 1

In this section, we test the ability of adaptive algorithms, namely, MAML, FOMAML and also the pre-trained model, to make predictions on the day 6 held out set by finetuning their parameters using only 252 trials. It is to be noted that multiple $Adaptation1$ finetuning data sets were tested using all combinations of frequency and amplitude. We observed that using the Adaptation set involving 600μA and 50Hz yields the best result for this held-out set, which was reported (refer Table 4.1). The results are evidence that pre-training performs better than both MAML and First-Order MAML for this problem setting, this suggests that the initialisation learnt by pre-training is more suitable for fine-tuning. However, we expected the contrary, since pre-training does not guarantee an initialisation that is good for fine-tuning. However, MAML and related algorithms directly optimise performance with respect to initialisation suitable for fine-tuning.

4.1.3 Adaptation 2

For the pre-trained model, fine-tuning model parameters using the $Adaptation2$ set leads to decreasing model performance, as compared to its initial parameters (refer Table 4.2). We attribute this drop in performance to over-fitting. Since this adaptation set contains only 36 trials, the fine-tuning process does not guide this model to a generalized solution. However, even though the L1 loss for adaptive models (MAML, FOMAML) trained using meta-learning

Model	Training Data	Finetuning Data	L1 Error (On held-out set from Day 6)
MLP	540 Trials from Day 6	-	0.0412
Pre-Trained MLP	All Trials from Day 1 to Day 5	-	0.0512
		36 Trials (Adaptation 2) from Day 6	**0.0762**
MAML	All Trials from Day 1 to Day 5	-	0.2315
		36 Trials (Adaptation 2) from Day 6	**0.1023**
First-Order MAML	All Trials from Day 1 to Day 5	-	0.2153
		36 Trials (Adaptation 2) from Day 6	**0.0893**

TABLE 4.2: Comparison of the performance of the baselines and the meta-learning algorithms (after fine-tuning using Adaptation 2) set on the held-out set from Day 6.

is higher, we see improvement by fine-tuning using this small adaptation set. This is because meta-learning guides the model to learn initialization which is optimized for fine-tuning. However, these methods do not perform at par or outperform the error by baseline MLP (0.0412). We discuss possible reasons and improvements for this in the next section.

4.2 Summary and Limitations

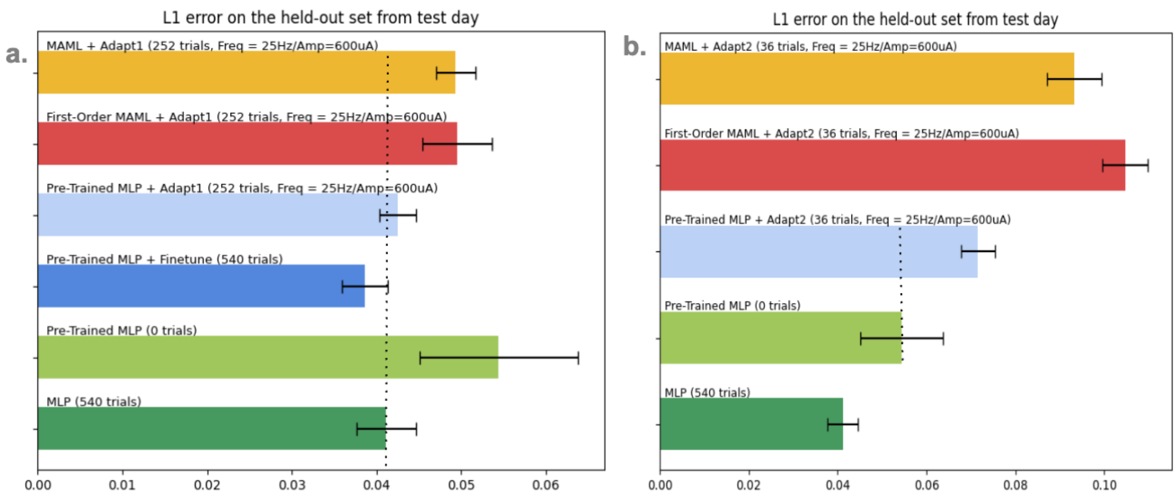

FIGURE 4.1: The results are summarized after cross-validation. We cross-validate the results by using different days from day 1 to 6 as the test day and by training the adaptive algorithms on the remaining 5 days. We observe that the trends of the results are consistent with Table 4.1 & Table 4.2 (results on Day 6) when tested every other day (Day 1 to Day 5).

Pre-training aids model performance, and even performs at par with a model trained using all trials from a day even with only half the number of trials. However, contrary to our expectation meta-learning-based models (MAML, FOMAML) failed to outperform pre-training. However, with this problem setting and this scale of loss, the difference is not very evident. We suggest further experiments as future work, mainly extending this work for kinematics predictions, to better understand the efficacy of these methods. It is interesting to note that fine-tuning using the Adaption 2 strategy led to deteriorating model performance for the pre-trained model which can be attributed to overfitting. However, they do help guide the model parameters to a better solution from the initialisation set by meta-learning (for MAML, FOMAML). But it does not outperform the baselines. The optimisation procedure for MAML involves computing double derivatives, hence, it is often not ideal for problems where we need to perform a large number of adaptation steps or gradient descent for learning the individual tasks. Hence, we discuss some possible improvements which will be attempted in future work.

Chapter 5

Conclusion and Future Direction

The success of a pre-trained model to predict the average EMG power suggests that it is possible to do better than re-training the forward model every day as part of our efforts to build a deployable BCI for SCI patients. One way to do this is by creating models (using meta-learning) with better model parameter initialisation.

The results in this thesis conclude that MAML and FOMAML do not perform at par with pre-training when it comes to fine-tuning (or re-calibrating) the trained model using a smaller number of trials. However, meta-learning seems like a promising approach as seen by the improvement of model performance while using the Adaptation 2 fine-tuning set for a model initialized using meta-learning. Although fine-tuning using this set led to over-fitting in the pre-trained model, it does lead to better performance for the meta-learned model. This will be further investigated.

Our results for the prediction of average EMG power need to be extended further to evaluate the true potential of these methods for the end goal of this project. In the future, we are aiming to build a forward model to directly predict the kinematics of motion elicited by the stimulations. We aim to test these meta-learning algorithms for the kinematics prediction task as immediate future work. It is to be noted that the algorithms examined in this thesis can be tested for any type of forward model since they are model agnostic. MAML was one of the preliminary works in the meta-learning literature and several improvements have been suggested since. One of the reasons for the failure of MAML in this problem setting could be the large number of adaptation steps (or gradient descent steps) required for this problem setting, since MAML optimization deals with double derivatives. This problem has been deeply studied in the literature, and several algorithms such as Reptile[25] have been proposed, which we are hopeful would yield better results. The investigation of these meta-learning techniques is left as future work.

Bibliography

[1] Harkirat Singh Behl, Atılım Güneş Baydin, and Philip HS Torr. "Alpha maml: Adaptive model-agnostic meta-learning". In: *arXiv preprint arXiv:1905.07435* (2019).

[2] Marco Bonizzato et al. "Autonomous optimization of neuroprosthetic stimulation parameters that drive the motor cortex and spinal cord outputs in rats and monkeys". In: *Cell Reports Medicine* (2023).

[3] Jonathan S Calvert et al. "Spatiotemporal distribution of electrically evoked spinal compound action potentials during spinal cord stimulation". In: *Neuromodulation: Technology at the Neural Interface* (2022).

[4] Kuan-Jung Chiang et al. "A closed-loop adaptive brain-computer interface framework: Improving the classifier with the use of error-related potentials". In: *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE. 2021, pp. 487–490.

[5] Henrique Dantas, V John Mathews, and David Warren. "Semi-supervised adaptive learning for decoding movement intent from electromyograms". In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE. 2019, pp. 1–5.

[6] Abdulbasit A Darem et al. "An adaptive behavioral-based incremental batch learning malware variants detection model using concept drift detection and sequential deep learning". In: *IEEE Access* 9 (2021), pp. 97180–97196.

[7] Alan D Degenhart et al. "Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity". In: *Nature biomedical engineering* 4.7 (2020), pp. 672–685.

[8] Alberto Diez-Olivan et al. "Adaptive dendritic cell-deep learning approach for industrial prognosis under changing conditions". In: *IEEE Transactions on Industrial Informatics* 17.11 (2021), pp. 7760–7770.

[9] Laura N Driscoll, Lea Duncker, and Christopher D Harvey. "Representational drift: Emerging theories for continual learning and experimental future directions". In: *Current Opinion in Neurobiology* 76 (2022), p. 102609.

[10] Laura N Driscoll et al. "Dynamic reorganization of neuronal activity patterns in parietal cortex". In: *Cell* 170.5 (2017), pp. 986–999.

[11] Mohammad Navid Fekri et al. "Deep learning for load forecasting with smart meter data: Online Adaptive Recurrent Neural Network". In: *Applied Energy* 282 (2021), p. 116177.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135.

[13] Marta Garnelo et al. "Conditional neural processes". In: *International conference on machine learning*. PMLR. 2018, pp. 1704–1713.

[14] Lakshmi Narasimhan Govindarajan et al. "Fast inference of spinal neuromodulation for motor control using amortized neural networks". In: *Journal of Neural Engineering* 19.5 (2022), p. 056037.

[15] Daniel Huber et al. "Multiple dynamic representations in the motor cortex during sensorimotor learning". In: *Nature* 484.7395 (2012), pp. 473–478.

[16] Brianna M Karpowicz et al. "Stabilizing brain-computer interfaces through alignment of latent dynamics". In: *bioRxiv* (2022), pp. 2022–04.

[17] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[18] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. 1. Lille. 2015.

[19] Denghao Li et al. "Model-agnostic meta-learning for EEG motor imagery decoding in brain-computer-interfacing". In: *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE. 2021, pp. 527–530.

[20] Zhenguo Li et al. "Meta-sgd: Learning to learn quickly for few-shot learning". In: *arXiv preprint arXiv:1707.09835* (2017).

[21] Nishtha Madaan et al. "DetAIL: A Tool to Automatically Detect and Analyze Drift In Language". In: *arXiv preprint arXiv:2211.04250* (2022).

[22] Paul Masset, Shanshan Qin, and Jacob A Zavatone-Veth. "Drifting neuronal representations: Bug or feature?" In: *Biological Cybernetics* 116.3 (2022), pp. 253–266.

[23] Alexander Mathis et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature Neuroscience* (2018). URL: `https://www.nature.com/articles/s41593-018-0209-y`.

[24] Tsendsuren Munkhdalai and Hong Yu. "Meta networks". In: *International conference on machine learning*. PMLR. 2017, pp. 2554–2563.

[25] Alex Nichol, Joshua Achiam, and John Schulman. "On first-order meta-learning algorithms". In: *arXiv preprint arXiv:1803.02999* (2018).

[26] Shanshan Qin et al. "Coordinated drift of receptive fields in Hebbian/anti-Hebbian network models during noisy representation learning". In: *Nature Neuroscience* (2023), pp. 1–11.

[27] Michael E Rule and Timothy O'Leary. "Self-healing codes: How stable neural populations can track continually reconfiguring neural representations". In: *Proceedings of the National Academy of Sciences* 119.7 (2022), e2106692119.

[28] Amal Saadallah and Katharina Morik. "Online ensemble aggregation using deep reinforcement learning for time series forecasting". In: *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. 2021, pp. 1–8.

[29] Adam Santoro et al. "Meta-learning with memory-augmented neural networks". In: *International conference on machine learning*. PMLR. 2016, pp. 1842–1850.

[30] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in neural information processing systems* 30 (2017).

[31] Flood Sung et al. "Learning to compare: Relation network for few-shot learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1199–1208.

[32] Sina Tafazoli et al. "Learning to control the brain through adaptive closed-loop patterned stimulation". In: *Journal of Neural Engineering* 17.5 (2020), p. 056007.

[33] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems* 29 (2016).