

IMPERIAL



IMPERIAL COLLEGE LONDON

DEPARTMENT OF BIOENGINEERING

**LINEAR AND NON-LINEAR METHODS OF SOURCE SEPARATION
IN NEUROPHYSIOLOGICAL TIME SERIES**

Alexander Kenneth Clarke

Supervised by Professor Dario Farina and Dr Paul Bentley

Submitted as part of the requirements for the degree of
Doctor of Philosophy in Bioengineering at Imperial College London
and the Diploma of Imperial College, June 2024

Statement of Originality

Except where otherwise stated, the work presented in this thesis is my own. The postscript at the beginning of each chapter states with whom the research was conducted and the extent of the collaboration.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Dedication

This thesis and the years of work it represents would not have happened without the patient encouragement and encouraging patience of a number of people in my life.

To my supervisor, Prof. Dario Farina, who from the first day trusted me to meander through my own path of interests, whilst always being there to listen and guide when needed.

To my co-supervisor, Dr Paul Bentley, who helped me greatly in my first year of trying to decide where to direct my studies.

To Dr Kate Hobson, who was always there to help me navigate the sometimes byzantine bureaucracy of Imperial College.

To Irene, Agne, Shihan, Pranav and the other members of the Prof. Farina's group, who have always been there to bounce ideas off and to commiserate with through endless major revisions.

To the team at Meta Reality Labs and in particular my supervisor Jorge, who made my time in New York such a fantastic experience.

To the camaraderie of my doctor and nurse colleagues at Queen Square, without which the pandemic would have been a completely overwhelming experience.

To the boundless happiness and support given to me by all my amazing friends in London.

To my brothers and grandparents, who have always tried to follow whilst I babbled about my research.

To my parents, who taught me that education was about giving one-self choices in life, and then were made to watch a seemingly endless series of career changes.

To Melissa, a beautiful captain, phenomenal sous-chef and terrible video-gamer.

To Lisa, my brown-eyed girl.

*For the thoughts are to the desires, as scouts and spies, to range
abroad, and find the way to the things desired.*

Thomas Hobbes
Leviathan

Abstract

Blind source separation of neurophysiological time series into its constituent neural activity is a key component of modern concepts of neural interfacing and an important tool for motor neuroscientists. With concurrent rapid advances in high-density electrode arrays, there is now a real possibility of devices driven directly by the nervous system leaving the lab and becoming a core part of machine interaction in everyday life. Such an advance would represent a leap forward in a diverse range of technologies, including intuitive prosthetics, neurological diagnostics, and consumer device. However, contemporary linear algorithms can return poor results when subjected to some of the signal contaminants that will be faced in the wild, such as high or varying noise levels and the non-stationary effects of dynamic contraction. New approaches are needed if source separation routines are to be robust enough for practical use.

One clear avenue of enquiry can be found in the recent explosion in techniques for training non-linear functions such as deep neural networks. Deep neural networks are highly flexible and have proven efficacy on a variety of data types, including many time series applications. However, their adoption would significantly increase the complexity of the blind source separation pipeline, so their presence would need to be well justified. In this thesis I explore a range of deep learning methodologies, and in particular how they can be effectively blended with contemporary linear algorithms. I demonstrate that such hybrid approaches can bring a range of advantages, such as improved management of noise, the automatic identification of incorrect timestamps and the ability to compensate for non-stationary effects in the signal.

Contents

Contents	vii
List of Figures	xi
1 Introduction	1
1.1 Outline of Thesis	3
1.2 A Brief History of EMG Decomposition	4
1.3 The Blind Source Separation Pipeline	5
1.3.1 Forward Model	7
1.3.2 EMG Recording	7
1.3.3 Noise and Signal Corruption in EMG	8
1.3.4 Extension	9
1.3.5 Whitening	11
1.3.6 Independent Component Analysis	12
1.3.7 Timestamping	14
1.3.8 Avoiding Redundant Projections	16
1.4 Deep Neural Networks	17
1.4.1 Neural Network Architectures	18
1.4.2 Optimisation	19
1.4.3 Backpropagation	19
2 Training Recurrent Neural Networks to Decompose Surface EMG into MU Activity¹	21
2.1 Introduction	23
2.2 Theory and Algorithms	25
2.2.1 Gradient Convolution Kernel Compensation	25
2.2.2 Gated Recurrent Unit Networks	27
2.2.3 Decomposition Accuracy	29
2.3 Materials and Methods	30
2.3.1 Simulated Data Generation and Decomposition	30
2.3.2 Experimental sEMG Dataset	31
2.3.3 Data Pipeline and Training	33
2.4 Results	36
2.4.1 Simulated sEMG Signals	36
2.4.2 Experimental sEMG Signals	37

2.5	Discussion	40
2.6	Conclusion	43
3	Pretraining Recurrent Neural Networks with Finite Element Method Simulations of EMG¹	45
3.1	Introduction	47
3.2	Methods	49
3.2.1	Forward Problem	49
3.2.2	EMG Signal of a Single Fiber	56
3.2.3	Modelling of Motor Units	58
3.2.4	Pretraining a Neural Network	59
3.2.5	Neural Network Architecture and Training	60
3.2.6	Experimental Validation	62
3.3	Results	63
3.3.1	Simulation Outputs	63
3.3.2	Pretraining Results	65
3.4	Discussion	67
4	Using Deep Metric Learning to Build Self-Correcting MU Decomposition Pipelines¹	73
4.1	Introduction	75
4.2	Related Work	79
4.3	Preliminaries	80
4.3.1	Deep Metric Learning	80
4.3.2	Locality Sensitive Mining	81
4.3.3	Angular Loss	82
4.3.4	Cross Entropy Loss	83
4.3.5	Neurophysiological Signal Decomposition	83
4.4	Validation Methodology	84
4.4.1	Evaluated Methods	84
4.4.2	Experiments	85
4.4.3	Model and Training	88
4.4.4	Experimental Dataset	90
4.5	Results	92
4.5.1	Feature-dependent Label Noise	92
4.5.2	Class-dependent Label Noise	93
4.5.3	Rediscovering Unlabelled Activations	95
4.5.4	Impact of the k Hyperparameter	96
4.6	Conclusion	97
5	Compensating for Non-Stationarities in EMG with Deep Learning¹	99
5.1	Introduction	101
5.2	Method	102
5.2.1	Preliminaries	102
5.2.2	Independent Component Analysis	105
5.2.3	Estimating a Nonstationarity	107

5.2.4	HarmonICA: A Quasilinear ICA Model	108
5.2.5	HarmonICA in Practice	110
5.3	Experiments	112
5.3.1	Real Intramuscular EMG with Synthetic Drift	112
5.3.2	Simulated Non-isometric Surface EMG	112
5.3.3	Real Non-isometric Surface EMG	113
5.3.4	Implementation Details	114
5.4	Results	116
5.4.1	Intramuscular EMG	116
5.4.2	Simulated Surface EMG	116
5.4.3	Experimental Surface EMG	116
5.5	Limitations	117
5.6	Conclusion	118
6	Conclusion	119
6.1	Summary	119
6.2	Future Perspectives	121
6.3	Final Comments	123
	Bibliography	125

List of Figures

1.3.1 Generation and Decomposition of EMG	6
2.2.1 Gated Recurrent Unit Architecture	28
2.3.1 Simulated Surface Electromyography Signal	30
2.3.2 Pipeline for Decomposition of Electromyography Signal with Deep Learning	32
2.3.3 Plot of GRU vs gCKC Performance on Simulated Data	34
2.4.1 Raster Plot Comparing GRU and gCKC Outputs	37
2.4.2 GRU Performance vs gCKC on Two Source Validation	38
2.4.3 GRU vs gCKC Performance for Different Levels of Additive Noise	40
3.1.1 Simulation Pipeline for Surface EMG	50
3.2.1 Simulated Muscle Geometry Model	52
3.2.2 Pretraining Pipeline Using FEM Simulation Outputs	61
3.3.1 FEM Simulation Examples	66
3.3.2 Results of Pretraining with FEM Simulation Outputs	68
4.1.1 Preserving Local Structures in Deep Metric Learning	78
4.4.1 The DeepDecomp Pipeline for Cleaning a Label Set	85
4.4.2 Effect of DML Loss on the Embedding Space	86
4.5.1 Principal Component Projection of Embedding Space	93
4.5.2 Demonstration of DeepDecomp on a Noisy IPT	94
4.5.3 Effect of the top-k Hyperparameter in LSM	97
5.2.1 HarmonICA in Operation on Dynamic sEMG	111

List of Abbreviations

- BSS: Blind Source Separation** - A technique to separate a set of source signals from a set of mixed signals without knowing the form of the sources.
- cBSS: convolutive Blind Source Separation** - An extension of gCKC that adds a final step of k-means source refinement.
- CE: Cross-Entropy** - A loss function used in machine learning to measure the difference between two probability distributions.
- CSD: Current Source Density** - The distribution of current sources within a volume conductor.
- DML: Deep Metric Learning** - A type of learning that focuses on learning a distance metric in a deep learning context.
- EMG: Electromyography** - A technique for evaluating and recording the electrical activity produced by muscles.
- FEM: Finite Element Method** - A numerical technique for finding approximate solutions to boundary value problems for partial differential equations.
- gCKC: gradient Convolution Kernel Compensation** - A EMG decomposition algorithm that uses gradient descent to refine an LMMSE estimator.
- GRU: Gated Recurrent Unit** - A type of recurrent neural network that can stably capture dependencies for time series.
- HMI: Human Machine Interface** - Interfaces that allow humans to interact with machines or computers.
- ICA: Independent Component Analysis** - A computational method to separate a multivariate signal into additive, independent components.
- IPT: Innervation Pulse Train** - An estimated sequence of motor neuron impulses sent to a muscle.
- JADE: Joint Approximation Diagonalization of Eigen-matrice** - An algorithm used for full independent component analysis.
- LMMSE: Linear Minimum Mean Square Error** - A statistical method for estimating the values of an unknown signal.
- LSTM: Long Short Term Memory** - A type of recurrent neural network that can stably learn long-term dependencies.
- MDR: Multi-level Distance Regularisation** - A regularization method used in DML.
- MU: Motor Unit** - The pool of muscle fibers innervated by a single motor neuron.

MUAP: Motor Unit Action Potential - The electrical signal produced by a motor unit during muscle contraction.

MVC: Maximum Voluntary Contraction - The greatest amount of force that a muscle group can exert voluntarily.

NL: N-pair Loss - A loss function used in DML to enhance the performance of the model by comparing multiple pairs.

NN: Neural Network - A computational model consisting of interconnected nodes.

PCA: Principal Component Analysis - A statistical procedure that uses orthogonal transformation to convert a set of correlated variables into a set of uncorrelated variables.

PNR: Pulse Noise Ratio - A quality measure of an IPT estimate.

RNN: Recurrent Neural Network - A family of neural networks where connections between nodes form a directed graph along a sequence.

RoA: Rate of Agreement - A metric used to evaluate the consistency or agreement between two sources of timestamps.

SNR: Signal-to-Noise Ratio - A decibel measure to quantify the level of a signal to the level of background noise.

STA: Spike-Triggered Average - A method to extract the average response of a MU activation.

SVD: Singular Value Decomposition - A mathematical technique used in linear algebra to decompose a matrix into its constituent elements.

ZCA: Zero Component Analysis - A technique for whitening data to make the components uncorrelated and have a unit variance.

Chapter 1

Introduction

Chapter Abstract

Tools which enable the decomposition of neurophysiological time series into constituent neural activity are vital to both neuroscientists seeking to understand the motor system and neuroengineers looking to build the next generation of human machine interfaces. This is particularly true in the domain of electromyography (EMG) recordings, where source separation gives direct access to spinal motor neuron activity. Better decomposition technologies in this domain have the potential to unlock a new paradigm of intuitive prosthetic limbs, discrete inputs to consumer hardware, rehabilitation systems after nervous system insults and even third limbs through motor augmentation. However, despite many successes in recovering motor activity from EMG under controlled conditions in the last 15 years, there remains a significant gap between the recordings that can be processed by current state-of-the-art algorithms and those possible under the full range of human motor dynamics. One potential avenue forward is to invoke the massive advances seen in the field of machine learning that have occurred in recent years. In this chapter we explore the theoretical basis of the classical and modern approaches that are explored in this thesis.

1.1 Outline of Thesis

The focus of this thesis can be broadly described as an exploration of how modern non-linear deep learning techniques can be used to augment classical linear source separation methods in electromyographic recordings. In the rest of Chapter 1, I will begin with a description of EMG decomposition algorithms leading up to the current state-of-the-art, before discussing relevant advances in machine learning.

In Chapter 2, I describe a supervised deep learning pipeline by which a recurrent neural network (RNN) can be used to identify a motor unit (MU) activity in surface EMG (sEMG) signal. Of particular note is the ability of the network to outperform the classical algorithm used to train it, a theme which will pervade much of the rest of this thesis,

In Chapter 3, I discuss a method of pretraining an RNN using MU activation potential (MUAP) waveforms from an advanced finite element method simulation of the sEMG forward model. This has the effect of significantly improving the efficiency of adaptation to a real sEMG decomposition.

In Chapter 4, I propose the use of a deep metric learning (DML) paradigm for identifying and classifying MUAPs in sEMG. This has a number of advantages over just learning an activation time series for each source. As an example, I show that the pipeline can be used to identify false positive and false negative MUAPs in timestamped activity, allowing classical sEMG decompositions to be “cleaned”.

In Chapter 5, I put forward a novel method of incorporating deep learning into linear source separation routines to stabilise the common non-stationarities seen in sEMG and intramuscular EMG (iEMG). I show this allows decomposition of sEMG recorded under non-isometric conditions and of iEMG contaminated by electrode drift.

Finally, in Chapter 6, I summarise the findings of the thesis and conclude. I will also use the opportunity to discuss future directions and challenges for the field of source separation in neurophysiological time series.

1.2 A Brief History of EMG Decomposition

Despite the many poorly-understood complexities of the human motor system, from persistent activity encoding in the prefrontal cortex, to incorporation of central pattern generation signals into spinal motor neurons, every interaction with our world is actuated through the humble motor unit activation. It is remarkable that all outputs of a system of such complexity can be described through a single method of encoding that is physically accessible through minimally-invasive and even non-invasive recording modalities. Only a final step of source separation stands between the spinal motor neuron activity and the operator or device wishing to utilise them. However, this has provided a considerable challenge for a variety of reasons we will detail in this section.

Early methods of EMG decomposition focused on recovering single MU activity from wire or needle iEMG[1, 2, 3], and at the turn of the millennium the base approach was to use second order statistics to source separate[4, 5, 6], i.e. template matching routines, usually followed by additional steps of collision management. This paradigm would be rapidly replaced over the next few years by the emergence of the high density electrode array and the rise in blind source separation (BSS) techniques using higher order statistics[7, 8, 9]. These approaches are far more resistant to complex superposition and so better suited to sEMG where selectivity is poor compared to iEMG.

BSS algorithms for EMG have generally fallen within the theoretical framework of inde-

pendent component analysis (ICA), where projections are designed to maximise higher order statistical moments, which can be interpreted as maximising the statistical independence of the sources in a signal[10]. Akazawa was the first to explore the application of this framework to EMG[7], using the FastICA projection pursuit algorithm[10], although their model did not account for the convolutional nature of the EMG forward process and so lead to poor outcomes. Similar results were found when they implemented a full ICA algorithm[11], in this case joint approximation diagonalization of eigen-matrices (JADE) algorithm[12]. The true potential of ICA did not become apparent until the introduction of gradient convolution kernel compensation (gCKC) by Holobar and Zazula[9, 13], which had the dual innovations of approximating an instantaneous mixture model and initialising a projection on a likely MUAP occurrence. An important update to this method was proposed by Negro *et al*[14], called convolutive blind source separation (cBSS), a final step of source refinement was added to maximise the interspike interval coefficient of variance. The cBSS algorithms remains the most commonly used BSS algorithm for decomposition of iEMG and sEMG recorded with high density electrode arrays.

1.3 The Blind Source Separation Pipeline

The core BSS framework for linear EMG decomposition has not seen major changes in the last ten years. After a brief examination of the forward model we will go through the basic elements, although a more detailed analysis is given where needed in each subsequent chapter.

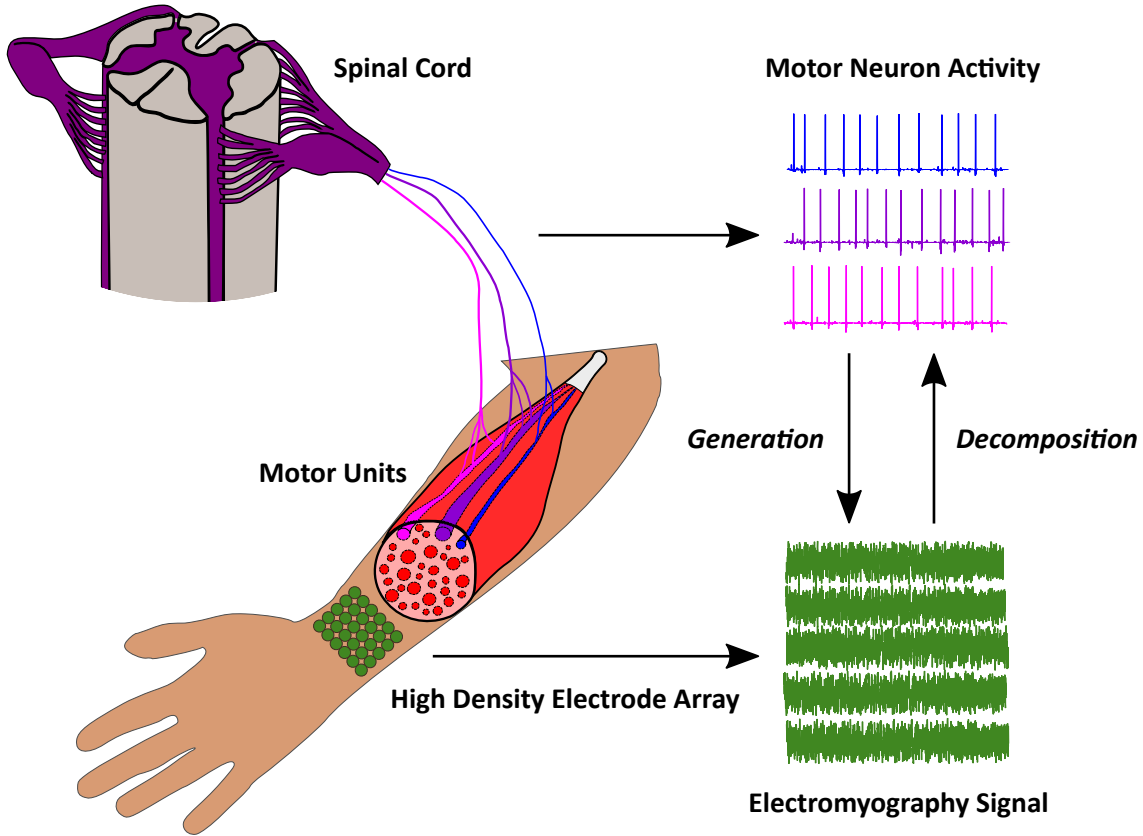


Figure 1.3.1: The forward model of EMG generation and the objective of blind source separation algorithms. Motor neurons project from the anterior horn of the spinal cord to connect with a pool of muscle fibers, forming a motor unit. When a motor neuron activates, it triggers an activation of the motor unit fibers at a one-to-one correspondence. The resultant motor unit activation potential traverses the volume conductor between the unit and the electrode, which has an attenuating effect on the waveform. Finally the outputs of many motor units are summed, creating a convolutional mixture. The objective of motor unit decomposition is to recover the motor neuron activity by using blind source separation techniques on the recorded EMG activity. The implied matrix inversion is made possible by the use of high density electrode arrays, which give a multivariate view of the volume conductor-attenuated motor unit activation potentials.

1.3.1 Forward Model

The forward myoelectric model of isometric EMG generation is well understood[15], and is illustrated in figure 1.3.1. Each motor neuron activation triggers a pool of associated muscle fibers, the motor unit, at a one-to-one correspondence[16]. The resultant MU action potential effectively acts as an amplifier of the motor neuron activity, making inference of motor neuron spiking possible even at some distance. However, the intervening volume conductor between the MU and the M electrodes introduces significant attenuation to the detected MUAP[4]. The electrodes detect a linear summation of all the MUAPs which have not been completely attenuated. We can model this mathematically as a convolutive mixture from a group of finite impulse response filters:

$$\mathbf{x}(t) = \sum_{l=0}^{L-1} \mathbf{H}(l) \mathbf{s}(t-l) + \boldsymbol{\omega}(t) \quad (1.3.1)$$

where the vector of sources $\mathbf{s}$ is a point process driven by N underlying neural inputs φ , i.e. the j -th source is defined by $s_j = \sum_r^R \delta(t - \varphi_r)$ where δ is the Dirac delta function and r is a discrete time instant contributing to the point process within the observed period. The mixing matrix $\mathbf{H}$ is size $m \times n$, containing a sample of MUAP templates for each observation m - source n combination, whilst $\boldsymbol{\omega}$ is a vector of additive Gaussian noise.

1.3.2 EMG Recording

Capturing both sEMG and iEMG signals involves a sequence of specific steps to ensure accuracy and reliability[17, 18]. For sEMG, initially a grid of electrodes is carefully placed

on the skin over the muscle of interest, with attention to electrode material, size, inter-electrode distance, and anatomical landmarks to optimize signal capture[19]. In contrast, iEMG involves the insertion of fine-wire or needle electrodes directly into the muscle, requiring detailed documentation of electrode type, insertion depth, and muscle region, often with the aid of ultrasound imaging[20, 17]. Once the electrodes are in place, the signals from both sEMG and iEMG are filtered to remove noise and artifacts, typically using band-pass filters to retain the relevant frequency components of the muscle activity[18]. The filtered signals are then amplified and digitized at high sampling rates to accurately capture the electrical activity of motor units[21]. Further digital filtering and resampling may be conducted based on the application specifications.

1.3.3 Noise and Signal Corruption in EMG

Electromyographic (EMG) signals are susceptible to various types of noise and corruption that can affect the accuracy and reliability of data analysis. One primary source of noise is power line interference, typically at 50 or 60 Hz, which arises from the electromagnetic fields of nearby electrical equipment and can significantly contaminate the signal. This interference is often mitigated through notch filters or adaptive filtering techniques[22]. Another common source is motion artifacts, caused by relative movement between the electrodes and skin, leading to low-frequency noise that distorts the true muscle activity signal. Proper electrode placement, secure fixation, and preprocessing techniques like high-pass filtering help address this issue[23].

Additionally, cross-talk from adjacent muscles can introduce unwanted components into the EMG signal, particularly in sEMG recordings. This occurs when the activity of neighbor-

ing muscles contributes to the detected signal, making it challenging to isolate the muscle of interest[24]. Advanced spatial filtering methods, such as high-density EMG arrays, can reduce cross-talk by leveraging the spatial distribution of the recorded signals[25]. Instrumentation noise, stemming from the inherent electronic noise of amplifiers and data acquisition systems, is another factor, often mitigated by ensuring high-quality equipment and appropriate signal amplification[26].

1.3.4 Extension

At this stage we have an EMG recording of shape $T \times M$ signal $\mathbf{x}$ consisting of T samples and M channels, which we then centre by subtracting the mean $E[\mathbf{x}]$ - a size M null vector. The objective is to invert the mixing matrix in equation 1.3.1 and so recover the motor neuron sources, however this is significantly complicated by the convolutional nature of mixing. Whilst there exists a number of convolutive BSS procedures, if we approximate local stationarity, then the convolutive mixture can be approximated as instantaneous by simply extending the observation and source matrices in time with the $L - 1$ previous samples[27]:

$$\tilde{\mathbf{x}}(t) = \tilde{\mathbf{H}}\tilde{\mathbf{s}}(t) + \tilde{\mathbf{w}}(t) \quad (1.3.2)$$

where the extended observation vector $\tilde{\mathbf{x}}$ can be written in block format as:

$$\begin{aligned} \tilde{\mathbf{x}}(t) &= [\tilde{\mathbf{x}}_1(t), \dots, \tilde{\mathbf{x}}_m(t), \dots, \tilde{\mathbf{x}}_M(t)]^T \\ \tilde{\mathbf{x}}_m(t) &= [x_m(t), \dots, x_m(t-l), \dots, x_m(t-L+1)] \end{aligned}$$

Chapter 1

and the extended source vector $\tilde{\mathbf{s}}$ as:

$$\tilde{\mathbf{s}}(t) = [\tilde{\mathbf{s}}_1(t), \dots, \tilde{\mathbf{s}}_n(t), \dots, \tilde{\mathbf{s}}_N(t)]^T$$

$$\tilde{\mathbf{s}}_n(t) = [s_n(t), \dots, s_n(t-l), \dots, s_n(t-L+1)]$$

and finally $\tilde{\mathbf{H}}$ is the extended mixing matrix:

$$\tilde{\mathbf{H}} = \begin{bmatrix} \tilde{\mathbf{h}}_{11} & \cdots & \tilde{\mathbf{h}}_{1n} & \cdots & \tilde{\mathbf{h}}_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{h}}_{m1} & \cdots & \tilde{\mathbf{h}}_{mn} & \cdots & \tilde{\mathbf{h}}_{mN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{h}}_{M1} & \cdots & \tilde{\mathbf{h}}_{Mn} & \cdots & \tilde{\mathbf{h}}_{MN} \end{bmatrix} \quad \tilde{\mathbf{h}}_{mn} = \begin{bmatrix} h_{mn}(0) & \cdots & h_{mn}(L-1) & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_{mn}(0) & \cdots & h_{mn}(L-1) \end{bmatrix}$$

The standard linear ICA model can then be used to invert $\mathbf{H}$ and so recover the unknown motor neuron sources[28]. In practice the true temporal support L of the MUAP is unknown, meaning the extension factor is somewhat arbitrary and is generally treated as a hyperparameter in the BSS model. Larger values tend to improve the stability of separation vector optimisation, at the cost of computational complexity.

The objective of inverting the mixing matrix gives us our full linear ICA model on the instantaneous approximation:

$$\tilde{\mathbf{s}} = \mathbf{W}\tilde{\mathbf{x}} \tag{1.3.3}$$

where the separation matrix $\mathbf{W}$ consists of the separation vectors $\mathbf{w}$, and the j -th vector $\mathbf{w}_j$

aims to invert the j -th column of $\tilde{\mathbf{H}}$ and so discover $\tilde{\mathbf{s}}_j$:

$$\tilde{\mathbf{s}}_j = \mathbf{w}_j^T \tilde{\mathbf{x}} \quad (1.3.4)$$

1.3.5 Whitening

The next preprocessing step is spatial decorrelation and normalisation of the extended observation matrix, where we find a whitening matrix $\mathbf{A}$ such that the second moment of the transformed extended observation matrix is the identity $\mathbf{I}$:

$$E[\tilde{\mathbf{x}} \mathbf{A} \mathbf{A}^T \tilde{\mathbf{x}}^T] = \mathbf{I} \quad (1.3.5)$$

given that $\mathbf{x}$ and hence $\tilde{\mathbf{x}}$ are zero-mean. Here the expectation is taken over all time instants, i.e. this is spatial whitening.

Whitening dramatically improves the optimisation speed and stability of ICA by effectively reducing the problem down to a rotation rather than rotation and scaling[10]. The effect of whitening can be seen by rewriting equation 1.3.2 to account for the new forward model with updated mixing matrix[29]:

$$\mathbf{A} \tilde{\mathbf{x}} = \mathbf{A} \mathbf{H} \tilde{\mathbf{s}} = \hat{\mathbf{H}} \tilde{\mathbf{s}} \quad (1.3.6)$$

Taking the second moment and equating with the identity:

$$E[\tilde{\mathbf{x}} \mathbf{A} \mathbf{A}^T \tilde{\mathbf{x}}^T] = \hat{\mathbf{H}} E[\tilde{\mathbf{s}} \tilde{\mathbf{s}}^T] \hat{\mathbf{H}}^T = \hat{\mathbf{H}} \hat{\mathbf{H}}^T = \mathbf{I} \quad (1.3.7)$$

Chapter 1

given that $E[\tilde{\mathbf{s}}\tilde{\mathbf{s}}^T]$ is unity as we assume independent and so uncorrelated sources. This means that the new mixing matrix is orthogonal and so only specified by around half the degrees of freedom of the original. This is also true for the separation matrix, greatly reducing the complexity of the problem.

There are a large number of methods for estimating $\mathbf{A}$, but by far the most popular in the literature are the principal component analysis (PCA) and zero component analysis (ZCA) methods. Both begin with either eigendecomposition or singular value decomposition (SVD) of the observation covariance matrix, we will focus here on an SVD of the extended observation variance:

$$E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \mathbf{V}\mathbf{S}\mathbf{U} \quad (1.3.8)$$

in which case the PCA whitening transform can be calculated by:

$$\mathbf{A}_{PCA} = \mathbf{S}^{-\frac{1}{2}}\mathbf{U}^T \quad (1.3.9)$$

and the ZCA transform as:

$$\mathbf{A}_{ZCA} = \mathbf{U}\mathbf{S}^{-\frac{1}{2}}\mathbf{U}^T \quad (1.3.10)$$

The additional rotation of ZCA has the effect of preserving the structure of the whitened version of $\tilde{\mathbf{x}}$, i.e. it has been sphered but rotation is left to the ICA step as much as possible[30].

For the rest of this chapter we will assume $\tilde{\mathbf{x}}$ represents its whitened form.

1.3.6 Independent Component Analysis

The actual process of ICA in the BSS pipeline can be roughly divided into full ICA, where all sources are estimated simultaneously, and projection pursuit ICA where sources are estimated

one-by-one. In the EMG literature the latter is by far the more popular[16]. Except for some reconstruction ICA formulations[31], full ICA generally requires that the separation matrix be square, which for extended multichannel EMG invokes a high computational burden. In either case, the independence assumption means that the ICA objective is applied individually to each estimated source. We can write this assumption as:

$$p_{\tilde{\mathbf{s}}}(\tilde{\mathbf{s}}) = \prod_{j=1}^M p_{\tilde{\mathbf{s}}_j}(\tilde{\mathbf{s}}_j) \quad (1.3.11)$$

The idea is to maximise the likelihood that the fitted generative model would produce the observed data, which through the change of variable rule and equation 1.3.3 can be written as:

$$p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) = p_{\tilde{\mathbf{s}}}(\mathbf{W}\tilde{\mathbf{x}}) |\det \mathbf{W}| \quad (1.3.12)$$

where $|\det \mathbf{W}|$ is the absolute determinant of the separation matrix and the output of the change of variable Jacobian. Its purpose is to ensure that the volume of $p(\tilde{\mathbf{x}})$ remains unity, which in optimisation routines mainly acts to prevent the weights blowing up during gradient ascent.

Using equation 1.3.11 this can be factored into:

$$p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) = |\det \mathbf{W}| \prod_{j=1}^M p_{\tilde{\mathbf{s}}_j}(\mathbf{w}_j^T \tilde{\mathbf{x}}) \quad (1.3.13)$$

and the ICA loss we seek to minimise is the negative likelihood:

$$\mathcal{L}(\mathbf{W}) = -|\det \mathbf{W}| \prod_{j=1}^M p_{\tilde{\mathbf{s}}_j}(\mathbf{w}_j^T \tilde{\mathbf{x}}) \quad (1.3.14)$$

Chapter 1

In the projection pursuit formulation the L1 norm of $\mathbf{W}$ is generally constrained to be unity during optimisation, which allows the determinant to be dropped. This allows a simple projection pursuit formulation, which in its estimated form can be written as:

$$\mathcal{L}(\mathbf{w}_j) = -E[p_{\tilde{\mathbf{s}}_j}(\mathbf{w}_j^T \tilde{\mathbf{x}})] \quad (1.3.15)$$

The exact form of $p_{\tilde{\mathbf{s}}_j}(\cdot)$ is unknown, however, we can make an excellent guess as to how it might look. We know spiking neurons form highly sparse distributions, so a function that is maximised by a supergaussian distribution is likely to optimise projections which highlight activity. We also know by the central limit theorem that a mixture of sources is likely to be more Gaussian than a single source, so optimising for supergaussianity is a good objective. Estimators which fulfil this objective include $p_{\tilde{\mathbf{s}}_j}(\cdot) \approx \log \cosh(\cdot)$ and $p_{\tilde{\mathbf{s}}_j}(\cdot) \approx \frac{1}{3}(\cdot)^3$ [14], and their selection is essentially a hyperparameter choice based on estimates of signal-to-noise ratios and desired convergence speeds.

With the final form of the loss function selected, the separation vector is then optimised to find the source estimates. Examples of both first and second order optimisation routines can be found in the EMG decomposition literature, as analytic formulas for the Jacobian and Hessian of $\mathcal{L}(\mathbf{w}_j)$ with respect to $\mathbf{w}_j$ can be found in advance. Gradient descent and fixed point iteration are by far the most commonly used optimisation algorithms.

1.3.7 Timestamping

Once the time series of a source, also called the innervation pulse trains (IPT) has been estimated, it needs to be converted into MU activation timestamps. A simple approach is to use a peak finding algorithm, but the time series will often contain some smaller spikes due

to noise or superposition. The most common method is to instead use k-means clustering with two centroids, with the selection threshold τ set to be the value half-way between the two centroids, i.e. hard assignment. A second step of separation vector optimisation can be added at this point[14], where $\mathbf{w}_j$ is replaced with the spike triggered average (STA) of the whitened observation vector. This process can be repeated, during which the predicted $\tilde{\mathbf{s}}_j(t)$ of time instants representing waveforms distant from that of the STA will be suppressed.

For a selected τ , the time instants of the activity time series can be split into those with spikes from the MU that the j -th separation vector is selective to, and the remaining null class:

$$\begin{aligned}\Psi_{spike} &\subseteq \{t \in T \mid \tilde{\mathbf{s}}_j(t) \geq \tau\} \\ \Psi_{null} &\subseteq \{t \in T \mid \tilde{\mathbf{s}}_j(t) < \tau\}\end{aligned}\tag{1.3.16}$$

We can calculate a quality statistic by comparing the values of $\tilde{\mathbf{s}}_j(\Psi_{spike})$ to $\tilde{\mathbf{s}}_j(\Psi_{null})$. The two most commonly used metrics are the pulse-to-noise ratio (PNR) and the silhouette measure. The PNR is a decibel ratio[32]:

$$\text{PNR} = 10 \log \frac{E[\tilde{\mathbf{s}}_j(\Psi_{spike})]}{E[\tilde{\mathbf{s}}_j(\Psi_{null})]}\tag{1.3.17}$$

whilst the silhouette is calculated using the centroid values from the k means clustering[14]. Every value $\tilde{\mathbf{s}}_j(t)$ is associated with its closest and furthest centroids, c_{in} and c_{out} , e.g. c_{in} and c_{out} for the time instants in Ψ_{spike} is the greater and lesser of the two centroids respectively.

The expectation of the normalised differences then give the silhouette metric: c_{spike} and c_{null} :

$$\text{Silhouette} = E \left[\frac{(c_{in} - \tilde{s}_j(t)) - (c_{out} - \tilde{s}_j(t))}{\max((c_{in} - \tilde{s}_j(t)), (c_{out} - \tilde{s}_j(t)))} \right] \quad (1.3.18)$$

The quality metrics can be used to decide whether to accept or reject a source, commonly threshold values of 30dB and 0.9 are used for PNR and silhouette respectively[32, 14].

If there is a comparative set of time stamps, for example the ground truth timestamps in a simulation or paired iEMG - sEMG timestamps in two source validation routines[33], a supervised statistic can be calculated. The rate of agreement (RoA) is found as follows:

$$RoA = \frac{M_{12}}{M_{12} + U_1 + U_2} \% \quad (1.3.19)$$

where M_{12} is the number of matched timestamps between the two methods, whilst U_1 and U_2 are the number of unmatched timestamps from the first and second decomposition method respectively. Usually a small tolerance of 0.5-5ms is used when deciding if a timestamp is temporally close enough to be considered a match.

1.3.8 Avoiding Redundant Projections

An important final consideration in projection pursuit ICA is the problem of converging on redundant sources, i.e. sources that have already been discovered. There are two main methods used to prevent this occurring, source deflation and peel off. In source deflation, the updated separation vector is orthogonalised to the column matrix of previously found

separation vectors on each optimisation step[10]:

$$\mathbf{w}_j^{(1)} = \mathbf{w}_j^{(0)} - \mathbf{W}\mathbf{W}^T\mathbf{w}_j^{(0)} \quad (1.3.20)$$

Peel off instead occurs once a source has been accepted. Its contribution to the observation matrix is first estimated by taking a STA with the output timestamps and then convolving that STA with the timestamps embedded into a length T binary time series. This contribution is then subtracted from the observation vector, before the next iteration of projection pursuit takes place on the residual.

Both methods can cause problems. Source deflation often fails to prevent redundant optimisations in highly multivariate time series[34]. Peel off will progressively “damage” the residual due to inaccuracies in source contribution estimation. In practice both methods continue to feature in the literature[16].

1.4 Deep Neural Networks

Deep neural networks are universal function approximators[35], meaning they can approximate any continuous function to a desired level of accuracy. They are able to learn complex non-linear relationships between inputs and outputs and are easily adapted to many different data types. In this section we will go through some of the basic theory, with more specific applications introduced in the following chapters as needed.

1.4.1 Neural Network Architectures

The structure of neural networks has become highly heterogeneous as the field has developed. For simplicity of explanation we will focus on the most basic, a feed-forward network. Feed-forward networks are constructed from layers, which are affine transformations followed by a non-linearity σ , called the activation function, whose purpose is to allow the network to efficiently model non-linearities:

$$\mathbf{z}^{(k)} = \sigma(\boldsymbol{\beta}^{(k)} \mathbf{z}^{(k-1)} + \boldsymbol{\epsilon}^{(k)}) \quad (1.4.1)$$

where k is the number of layers of the neural network, $\mathbf{z}^{(k-1)}$ is a size F vector, $\mathbf{z}^{(k)}$ is a size G vector. $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\epsilon}^{(k)}$ are respectively a size $G \times F$ weight matrix and size G bias vector.

Common activation functions include the sigmoid, hyperbolic tangent and rectified linear unit:

$$\begin{aligned} \text{Sigmoid}(\mathbf{z}) &= \frac{1}{1 + e^{-\mathbf{z}}} \\ \text{Tanh}(\mathbf{z}) &= \frac{e^{\mathbf{z}} - e^{-\mathbf{z}}}{e^{\mathbf{z}} + e^{-\mathbf{z}}} \\ \text{ReLU}(\mathbf{z}) &= \max(0, \mathbf{z}) \end{aligned}$$

A simple three layer feed-forward network will then look like:

$$\begin{aligned} \text{Input layer: } \mathbf{z}^{(0)} &= \tilde{\mathbf{x}} \\ \text{Hidden layer: } \mathbf{z}^{(1)} &= \sigma(\boldsymbol{\beta}^{(1)} \mathbf{z}^{(0)} + \boldsymbol{\epsilon}^{(1)}) \\ \text{Output layer: } \mathbf{z}^{(2)} &= \sigma(\boldsymbol{\beta}^{(2)} \mathbf{z}^{(1)} + \boldsymbol{\epsilon}^{(2)}) \end{aligned}$$

In practice, deep neural network architectures are far more complex, with specialised architectures depending on the task. These specialisations introduce inductive biases into the model that improves the generalisation performance for a given amount of data[36], for example convolutional neural networks are useful for computer vision where local similarity is common, whilst RNNs like gated recurrent units (GRU) are effective at learning time series. There are also layer augmentations that improve optimisation and generalisation performance, such as dropout and batch normalisation[37, 38].

1.4.2 Optimisation

Deep neural networks are typically optimised to minimise a loss objective through first order methods, as the Hessian quickly becomes impractical to compute as the architecture increases in size. Stochastic gradient descent is the most common approach, usually augmented by adaptive approaches that modify step size based on historical gradients[39, 40]. Commonly an early stopping framework is used, which means a set of validation data is split out of the training data and then used to halt training when over-fitting is detected.

1.4.3 Backpropagation

Backpropagation is an algorithm used to efficiently compute the gradients of the loss with respect to the parameters at each network layer[41]. After forward propagation to get the loss function value, the gradients are computed using the chain rule of calculus. Going back to the three-layer feed-forward neural network, the gradient of the loss with respect to the weights $\beta^{(2)}$ is given by:

$$\frac{\partial \mathcal{L}}{\partial \beta^{(2)}} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(2)}} \sigma'(\mathbf{z}^{(2)}) \mathbf{z}^{(1)\top} \quad (1.4.2)$$

Chapter 1

where $\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(2)}}$ represents the gradient of the loss with respect to the output of the second layer and σ' the first derivative of the activation function.

Similarly, for the gradient with respect to the weights $\boldsymbol{\beta}^{(1)}$, we have:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\beta}^{(1)}} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(2)}} \sigma'(\mathbf{z}^{(2)}) \boldsymbol{\beta}^{(2)} \sigma'(\mathbf{z}^{(1)}) \right) \mathbf{z}^{(0)\top} \quad (1.4.3)$$

This recursive application of the chain rule allows the gradient to be propagated backward through the network, layer by layer. The parameters are then updated using the computed gradients. Modern autograd algorithms are extremely efficient at quickly finding the gradient of the loss with respect to to each parameter in the architecture[42].

Chapter 2

Training Recurrent Neural Networks to Decompose Surface EMG into MU Activity¹

¹This chapter was originally published as *Deep Learning for Robust Decomposition of High-Density Surface EMG Signals*[43] in *IEEE Transactions on Biomedical Engineering* (2020) with co-authors Seyed Farokh Atashzar^{2, 3}, Alessandro Del Vecchio², Deren Barsakcioglu², Silvia Muceli⁴, Paul Bentley⁵, Filip Urh⁶, Ales Holobar⁶ and Dario Farina². Some alterations have been made to better suit the overall thesis structure, including some editing down of the theory section where it duplicates the previous chapter. All included images are from the original paper.

² Department of Bioengineering, Imperial College London, UK

³ Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, USA

⁴ Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden

⁵ Department of Brain Sciences, Imperial College London, UK

⁶ Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

Chapter Abstract

Blind source separation algorithms, such as gradient convolution kernel compensation, can efficiently and accurately decompose high-density surface electromyography signals into constituent motor unit action potential trains. Once the separation matrix is blindly estimated on a signal interval, it is also possible to apply the same matrix to subsequent signal segments. Nonetheless, the trained separation matrices are sub-optimal in noisy conditions and require that incoming data undergo computationally expensive whitening. One unexplored alternative is to instead use the paired sEMG signal and BSS output to train a model to predict MU activations within a supervised learning framework. A gated recurrent unit network was trained to decompose both simulated and experimental unwhitened sEMG signal using the output of the gCKC algorithm. The results on the experimental data were validated by comparison with the decomposition of concurrently recorded intramuscular EMG signals. The GRU network outperformed gCKC at low signal-to-noise ratios, proving superior performance in generalising to new data.

2.1 Introduction

Recent advances in the non-invasive identification of motor unit firings by processing surface electromyographic signals are offering new tools to neuroscientists[44][45], as well as the possibility to establish accurate and intuitive human-machine interfaces (HMI)[46][47]. These applications have been enabled by multivariate blind source separation algorithms that exploit the spatio-temporal information of high density sEMG recordings[14][13]. These methods have proved highly effective at decomposing sEMG signal into relatively large numbers of MU spike trains[48].

At the core of the majority of contemporary sEMG decomposition algorithms is the numerical estimation of spatio-temporal separation vectors, which extract innervation pulse trains when applied to the observations. These are sparse time-series estimates of MU activity with the same sampling rate as the sEMG signal, which can then be thresholded into MU spike timestamps[9]. Fixed-point iteration[28] or gradient descent[9] are used to update the separation vectors based on a contrast function[10], which is empirically selected to maximise the sparsity of the output IPTs[16]. This procedure is effective even in the case of highly correlated sources[16].

Gradient convolution kernel compensation, an alternative approach developed by Holobar and Zazula[9][13], approximates IPTs using a linear minimum mean square error (LMMSE) estimator framework. The segments of the signal likely to contain MU action potentials are identified by the thresholding of an activity index, before being cross-correlated through the observation matrix to find similar MUAPs. In gCKC this cross-correlation vector is then optimised by gradient descent using similar contrast functions to those used in the fast independent component analysis algorithm[27]. More recent algorithms have combined direct

and indirect separation vector estimation, using FastICA to generate a separation vector and then tuning it by iteratively identifying maximally similar MUAPs[14].

Once identified, separation vectors can be applied to new data segments to extract MUs with low computational cost for online applications[46]. This approach is however limited by non-stationarity in the signal, for example heteroscedasticity due to variable contraction levels or sweat[49]. Moreover, with these approaches, the vector of sEMG signal at each time instant must be extended with delayed signals to transform the convolutive mixture to a linear instantaneous mixture[14]. The observations are also spatially decorrelated by whitening transform prior to decomposition to improve the convergence properties of the optimisation routine[10]. These additional preprocessing steps reduce computational complexity of gradient descent or fixed-point algorithms, but complicate their online application to newly recorded sEMG signal, causing delays of over 300ms, which may not be acceptable in HMI applications[50][51]. Moreover, whitening is also likely to increase the sensitivity of the system to noise[10]. To date no attempt has been made to introduce a fundamentally different approach to the data pipeline between the offline training and online application phases of a real-time sEMG decomposition.

Rather than directly using the trained separation vectors on new data, we hypothesised that a faster and more stable approach is to train a supervised algorithm in the offline phase, which can then decompose sEMG signal online. In this formulation, the IPT of each MU from the decomposed training signal is treated as a class of time-ordered labels, giving examples of successful decomposition. The supervised algorithm then operates on the unwhitened sEMG signal, learning how to denoise and separate the observation signals into individual MU activations for thresholding. A further advantage offered is the possibility to augment

the training data during learning, such as the addition of noise to promote robustness. One candidate for such sequence-to-sequence learning is the recurrent neural network, a type of artificial neural network specifically designed to take advantage of temporal dependencies in ordered data such as time series[52]. Modern RNNs such as gated recurrent units and long short-term memory (LSTM) are able to learn long and short-term temporal dependencies without the optimisation problems associated with earlier RNN designs[52][53]. These gated architectures have shown state-of-the-art performance on sequence-to-sequence translation of time series data[54][55]. GRU and LSTM networks have also recently been applied in gesture classification tasks using global sEMG inputs[56][57].

In this study, we outline a framework for the supervised decomposition of sEMG signal into IPTs by RNN, specifically using a GRU network trained on the time series output of the gCKC decomposition algorithm[13]. The algorithm was designed to maintain a strict separation of output classes, prioritising learned sources over generalising to new MUs. The accuracy of this decomposition method is demonstrated on both simulated data and experimental sEMG signals matched with sources found by decomposition of simultaneously-collected intramuscular EMG.

2.2 Theory and Algorithms

2.2.1 Gradient Convolution Kernel Compensation

The gCKC algorithm seeks to recover the underlying MU activities in the sEMG signal by inverting a convolutional mixture of finite impulse response filters, representing the temporal dispersion caused by volume conduction effects and the different distances between individual

Chapter 2

MU fibers and surface electrodes[4]. Inversion of the extended mixing matrix $\tilde{\mathbf{H}}$ gives the separation vectors with which the MU sources can be recovered. In the case of convolution kernel compensation techniques, this is achieved indirectly using an initial linear minimum mean square error estimate of the j -th source:

$$\hat{\mathbf{s}}_j(t) = \hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}^T \mathbf{C}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1} \tilde{\mathbf{x}}(t) \quad (2.2.1)$$

where $\hat{\mathbf{s}}_j(t)$ is the IPT of the estimated j th source at time t , $\hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}^T$ is the transposed cross-correlation vector between an activation of the j th source and the whitened extended observation matrix and $\mathbf{C}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}$ is the inverted autocorrelation matrix of the extended observation matrix. $\hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}$ is initialised with the time point that maximises the Mahalanobis distance calculated on the total sEMG input[13].

In the formulation of equation 2.2.1, $\hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}^T \mathbf{C}_{\tilde{\mathbf{x}}\tilde{\mathbf{x}}}^{-1}$ is a blind estimate of the j th column vector of the inverted extended mixing matrix[9]. In gCKC, $\hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}$, is then optimised by gradient descent, with the updated cross-correlation vector $\mathbf{c}_{\mathbf{s}_j\tilde{\mathbf{x}}}^\circ$ found by:

$$\mathbf{c}_{\mathbf{s}_j\tilde{\mathbf{x}}}^\circ = \mathbf{c}_{\mathbf{s}_j\tilde{\mathbf{x}}} - \alpha \sum_t \frac{\partial f(\hat{\mathbf{s}}_j(t))}{\partial \hat{\mathbf{s}}_j(t)} \tilde{\mathbf{x}}(t) \quad (2.2.2)$$

where α is the learning rate and $f(\cdot)$ is the differentiable cost function, designed to increase the sparsity of the inverted mixing matrix in a similar fashion to the contrast functions used in the fixed point FastICA algorithm[16]. After optimisation of $\hat{\mathbf{c}}_{\mathbf{s}_j\tilde{\mathbf{x}}}$ the estimated IPT can then be converted to timestamps by a thresholding procedure, parameterised by a two-class

K-means clustering algorithm[14].

2.2.2 Gated Recurrent Unit Networks

GRU networks were originally designed for statistical machine translation, but were quickly adapted to other ordered data such as time series[53]. This was in part due to their reduced parameterisation relative to the successful LSTM architecture, as well as better performance on small datasets[58]. The architecture of a gated recurrent unit is formulated as:

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}) \quad (2.2.3)$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}) \quad (2.2.4)$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t \otimes (W_{hn}h_{t-1} + b_{hn})) \quad (2.2.5)$$

$$h_t = (1 - z_t) \otimes n_t \otimes h_{t-1} \quad (2.2.6)$$

where x_t and h_{t-1} are respectively the input vector at time t and the hidden vector from the previous timestep. r_t , z_t and n_t are respectively the reset, update and new gate vectors. The output of the unit h_t is the new hidden vector, which is also the new output vector. The gates are parameterised by the weights W and biases b . σ and $\tanh$ are the sigmoid and hyperbolic tangent activation functions, whilst $\otimes$ is the Hadamard product. This is displayed schematically in fig. 2.2.1.

The cell can then be repeated in time and depth in a similar manner to a node in a simple feed-forward neural network. Layers above the input layer use the hidden output of the GRU layer below h_t as their input x_t . The parameters of z_t determine how temporal context is

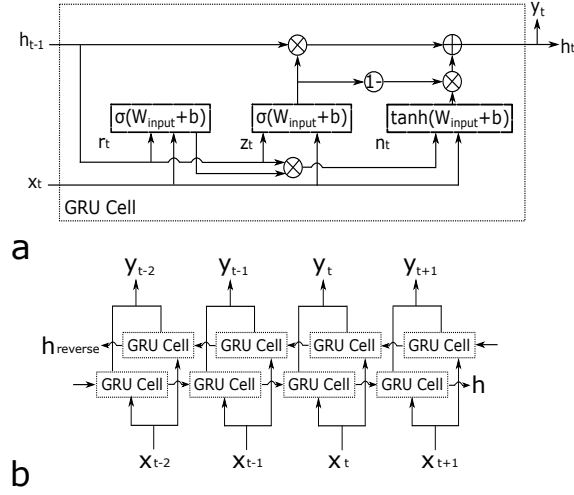


Figure 2.2.1: **a** shows the internal architecture of a gated recurrent unit. In **b** a single layer bidirectional gated recurrent unit network is shown. At each timepoint the output from both forward and backward cells are concatenated before being passed to the next layer of the network.

used to modify the processing of x_t , whilst the parameters of r_t determine what temporal information is relevant to future operations. If future observations are available, for example if data is processed in blocks, a bidirectional GRU architecture can be used (fig. 2.3.3).

To make class predictions, the weighted output of all cells in the top GRU layer are summed before an activation function is applied to make the final MU class predictions. The choice of this activation function has a major effect on network performance and convergence. In the context of IPT prediction, a MU activation is effectively a point process and is highly unlikely to co-occur with another source[13]. This property can be exploited by the softmax activation function:

$$Softmax(y_i) = \frac{\exp(y_j)}{\sum_j \exp(y_j)} \quad (2.2.7)$$

where each class prediction y_i is normalised by the summed exponents across all the classes. By constraining the class predictions in this way the separability of the output vector is maximised.

2.2.3 Decomposition Accuracy

A commonly used methodology to assess the accuracy of sEMG decomposition is the two-source validation technique, where the outputs of two separate decomposition methods operating on the same dataset are compared[24]. The reasoning for this is that a false MU activation is highly unlikely to be predicted by two independent procedures. The rate of agreement statistic was calculated between the output timestamps from the two methods.

In simulated data the ground truth of MU activation trains are known and are used as the second source for evaluation. In real sEMG data this is not available, instead the second source is generally a decomposed iEMG signal collected concurrently with the sEMG signal[24].

In the literature the thresholding level used to convert the IPT train to time stamps for RoA assessment is usually set by k-means clustering or by a multiple of the standard deviation calculated on the training set across all classes. The accuracy of this threshold selection has a major effect on decomposition accuracy beyond that of the algorithm generating the separation vectors. As the purpose of this study was instead to assess a separation procedure for maximisation of linear IPT separability, the thresholding level was set by a simple iterative procedure to maximise the calculated RoA for both the GRU and gCKC predictions.

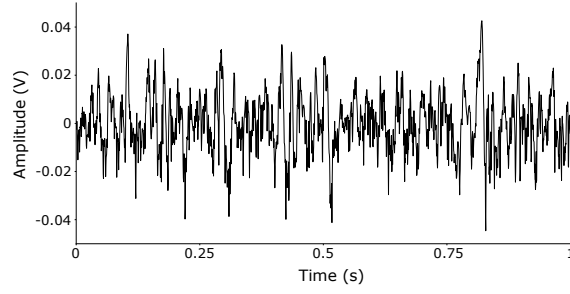


Figure 2.3.1: One second output of a single channel of the simulated sEMG signal, prior to the addition of noise. The signal consists almost entirely of complex superposition.

2.3 Materials and Methods

2.3.1 Simulated Data Generation and Decomposition

Synthetic MUAP templates as detected by a 24×8 channel sEMG matrix sampling at 2,048Hz with interelectrode distance 10mm were generated by simulation of muscle fibre recruitment in a 50 MU muscle modelled as a multilayered cylindrical volume conductor[59]. Four layers of thickness 7.5, 27.7, 2, and 1 mm were simulated to describe the bone, muscle tissue, fat, and skin layers respectively. The model included 101,276 individual muscle fibers (average semi-length 100 mm) with innervation zones distributed in a 10 mm region around half of the fiber length. The muscle was more conductive in the longitudinal than in the transversal fiber direction, at a conductivity ratio of 5:1. The conductivity ratios skin/fat layers and fat/muscles were respectively 20 and 0.5[60]. Fibers were assigned to local MU pools to give an exponential profile of unit size[61].

MU activations were generated as Poisson spike trains (20Hz) with average interspike intervals randomly selected between 30 and 40ms, simulating a constant force, with every MU active throughout the signal. Each MU spike train was then convolved with a simulated MUAP template, before summation to generate noiseless sEMG signals. These simulation

parameters resulted in sEMG signal which consisted primarily of complex MUAP superpositions (fig. 2.3.1). Seven 1-min sequences of sEMG signal were generated, with the only difference between them being the power of the additive normally-distributed noise, giving signal-to-noise ratios (SNR) of 30, 25, 20, 15, 10, 5 and 0dB.

Each simulated dataset was divided into a 40s long training set, a 10s long validation set and a 10s long testing set. The DEMUSE software-package[9] was used to decompose the training data into IPTs using gCKC (an example is shown in Fig. 2.3.2), with a cut-off pulse-to-noise ratio (PNR) of 30dB used as a threshold for source acceptance[32]. In order to evaluate the performance of a trained decomposition model on noisier unseen data a second decomposition was also performed using the separation vectors learned on the 30dB SNR dataset, which were then applied to the 25, 20, 15, 10, 5 and 0dB SNR datasets. For each of these SNRs none of the output IPTs were excluded on the basis of their calculated PNRs, meaning every dataset had the same number of sources.

2.3.2 Experimental sEMG Dataset

The performance of the GRU network was also analysed using a set of concurrently recorded sEMG and iEMG signals from the dominant tibialis anterior (TA) muscle of 12 men, a dataset previously used to validate the CKC algorithm[62]. A monopolar 12×5 electrode array with an interelectrode spacing of 5mm was placed over the main muscle innervation zone, with columns orientated to run parallel with the muscle fibers. The sEMG signal was band-pass filtered (10-500 Hz) and then sampled at 2,048Hz. Simultaneously the signals from a pair of wire electrodes inserted proximal to the sEMG matrix were band-pass filtered (500-5,000Hz) and sampled at 10,000Hz.

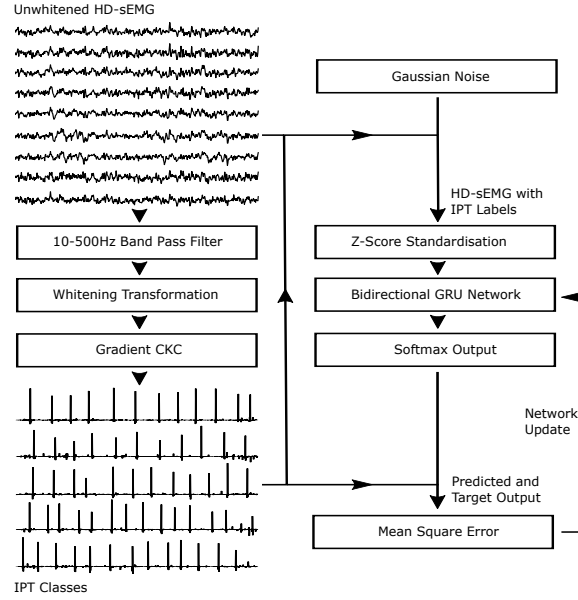


Figure 2.3.2: Algorithm pipeline during GRU network training. The IPT labels are generated by the unsupervised gCKC algorithm, which are then paired with the unwhitened and unfiltered sEMG signal for supervised training. The sEMG signal is augmented with Gaussian noise and z-score standardised. The mean square error between the softmax output and the IPT labels is then backpropagated through the GRU network.

Muscles forces were recorded by load cells mounted in isometric braces and sampled at 2048Hz. The maximum voluntary contraction (MVC) force was calculated as the the greatest force level expressed in three 5s maximal contractions separated by 3mins of rest. Subjects were then asked to perform two 20s non-consecutive isometric contractions at 5%, 10%, 15% and 20% MVC, with feedback provided by an oscilloscope. The first 12s of each experimental sEMG recording were selected as a training set, with the remaining 8s divided equally into 4s validation and test sets. The training set was then decomposed by gCKC into IPTs using the DEMUSE software package, with sources accepted if their calculated PNR was over 30dB. The trained separation vectors were then applied to the validation and test sets. The iEMG signals were decomposed into sources using EMGLAB[5]. Sources in the training set sEMG signal were then matched with sources found in the iEMG to a 5ms tolerance, with the

decomposition of a particular recording accepted if at least one source had a RoA of 70%. Recordings that did not meet this requirement were discarded.

For each accepted sEMG recording the GRU was trained on the 12s training set using the IPTs as labels. Predictions were then made by the trained GRU network on both the training set and unseen test set. The RoA was then calculated between the predictions and the gCKC output, again using a threshold that maximised this value. Where a source had an iEMG match, the maximum RoA was calculated between the iEMG-identified source and both the GRU network and gCKC separation vectors for both training and test sets.

To further test the robustness of the GRU to online heteroscedastic changes, an additional dataset was created by adding white Gaussian noise to the test set. Gaussian-distributed noise was selected as it is a common contaminant of sEMG signal and usually difficult to eliminate[63]. Signal amplitude was defined as the mean across all units in a recording of the maximum MU action potential amplitude across all channels, as found by spike-triggered averaging using the gCKC labels. Noise was then added to the test set to give SNRs of 30, 20, 10 and 0dB. The trained gCKC separation matrix was then applied to these test sets to give IPTs for comparison against GRU performance.

2.3.3 Data Pipeline and Training

The hardware used for both training and predictions was an Intel Core i7 7800X processor with a Geforce RTX 2080 Ti GPU. A schematic of the data pipeline is shown in fig. 2.3.2. The Pytorch machine learning package was used to build the GRU network, whilst Scikit-learn was used for data standardisation. Input sEMG data was first standardised by z-scoring based on the training set means and standard deviations, before being passed to the bidirectional GRU

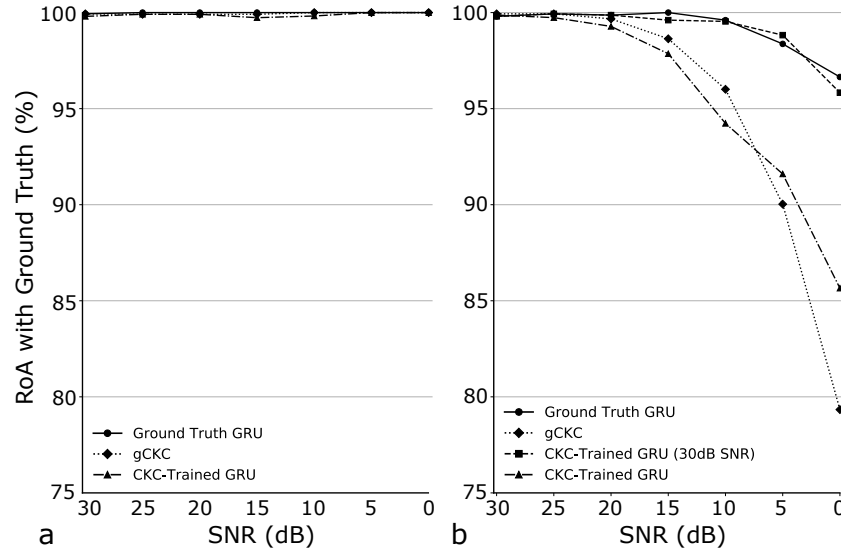


Figure 2.3.3: Performance of the GRU and gCKC algorithms on unseen simulated data at different SNRs as measured by RoA against the ground truth labels. In **a** only MU activation trains found by gCKC with a PNR of over 30dB were accepted. In **b** the separation filters calculated on the 30dB SNR training data was applied to training data of all SNRs. The GRU network was trained on ground truth labels, the 30dB SNR IPT train and the IPT generated from the vectors operating on the lower SNR training data.

network. The output of the GRU network was then passed to a linear node and softmaxed to give class predictions. The width and depth of the GRU network are hyperparameters and were optimised by grid-search on the training and validation sets of the 30dB SNR simulated data. Optimal GRU architecture was found to be an 80-cell wide network with two-layers. The Adam optimisation algorithm at a learning rate of 0.001, betas of 0.9 and 0.999 and epsilon of 10^{-8} was used for training. A 50% dropout probability on all GRU cells was used to prevent the network overfitting on the training set, with the additional theoretical advantage of removing overdependence on particular channels, improving robustness to electrode failure. A mean square error loss function was used to generate the error signal for backpropagation. Prior to training 10,000 windows of sEMG signal were randomly selected, each 400 samples wide, with a minibatch of 512 samples per update used. Data augmentation was conducted

during training by adding normally distributed noise of zero-mean and unit variance to each window, to encourage both robustness and to provide additional regularisation. Training was run based on an early-stopping framework, selecting an iteration that maximised a z-score based blind separability estimate $\bar{\epsilon}$ operating on the validation set GRU output y :

$$\bar{\epsilon} = \frac{1}{J} \sum_J \epsilon_j \quad (2.3.1)$$

where J is the total number of accepted MU sources and ϵ_j is given by:

$$\epsilon_j = (max_{0.25}(y_j) - \mu_{y_j}) / \sigma_{y_j} \quad (2.3.2)$$

where μ_{y_j} and σ_{y_j} are respectively the mean and standard deviations of the j th class output and $max_{0.25}(\cdot)$ is the mean of the maximal 0.25% of samples in a predicted IPT. This percentage is a training hyperparameter, with 0.25% found empirically to select for maximally sparse outputs whilst retaining robustness to signal noise. The same training and network hyperparameters were used for all simulated and experimental datasets.

For each dataset the GRU was trained on the unwhitened and unfiltered training sEMG signal. In the first simulated dataset, where the number of accepted MU trains was set by the PNR of gCKC operating at that SNR, the GRU was trained with both ground truth MU activation labels and on the output gCKC IPT. In the second simulated dataset, where the separation vectors learned on the 30dB SNR signal were used, the GRU was trained with both ground truth MU activation labels and on the output gCKC IPT as before, but additionally it was trained on the IPT output of the trained separation matrix from the 30dB

SNR decomposition. As with the gCKC output, the GRU class predictions on the test set were converted to time stamps by an iterative method of threshold parameter selection which maximised the RoA. This thresholding procedure was also used for the predictions made on the experimental data.

2.4 Results

2.4.1 Simulated sEMG Signals

In the first simulation dataset, where only MU trains found by the gCKC algorithm with PNRs above 30dB were accepted, performance on the unseen test data was close to 100% for the gCKC algorithm at all SNRs investigated (fig. 2.3.3a). However, the number of accepted MUs was much smaller in the low-SNR data (table 2.1). 11 MUs were identified in the 30dB SNR dataset, but this dropped rapidly as SNR increased. The high decomposition accuracy of the gCKC were mirrored by both the GRU network trained on the ground truth and the GRU network trained on the gCKC IPT.

More interesting results were obtained from the second dataset, where the separation vectors calculated on the 11 MUs found by the gCKC in the 30dB SNR set were applied to all other SNR levels (fig. 2.3.3b). The performance of these separation vectors declined rapidly as the SNR fell below 10dB. However, the GRU network trained on the IPT output from the 30dB SNR set performed much more robustly, closely matching the performance of the GRU trained directly on the ground truth. Even at an SNR of 0dB the GRU predictions retained a more than 95% RoA with the ground truth. Furthermore, when the gCKC decompositions of low SNR signal were used to train the GRU, the GRU went on to outperform the gCKC

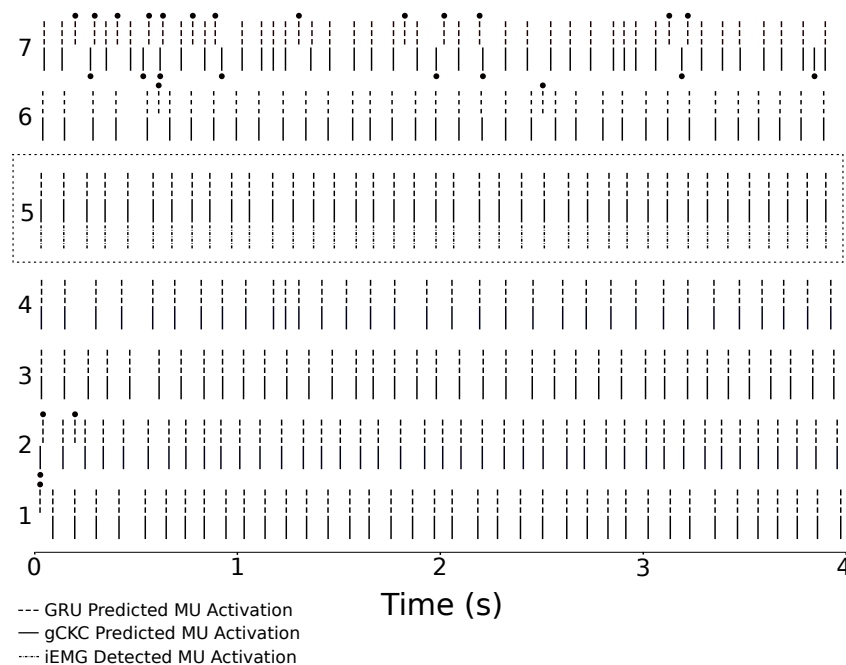


Figure 2.4.1: Raster plot showing both GRU and gCKC decompositions of 4s unseen sEMG signal from an isometric 10% MVC TA contraction. The MU activation train of unit 5 was matched with a source from the iEMG decomposition, which is also displayed. Disagreements between GRU and gCKC predictions are highlighted with dots and are confined mainly to unit 7. A 100% RoA is achieved between both the GRU and gCKC predictions and the iEMG decomposition for unit 5.

predictions on the unseen test data. This implies that the GRU is better able to generalise to new data than the unsupervised algorithm being used to train it, indicating that the GRU could learn the decomposition process even from imperfect labeling of the training data, improving on the performance of the gCKC algorithm used for labeling.

2.4.2 Experimental sEMG Signals

Twenty-six experimental recordings met the acceptance criteria of containing at least one identified source matched between the iEMG and sEMG signals to an RoA of at least 70%.

Table 2.1: Motor unit yield from gCKC decomposition at different signal-to-noise ratios.

SNR (dB)	Number of MUs
30	11
25	8
20	8
15	8
10	4
5	2
0	1

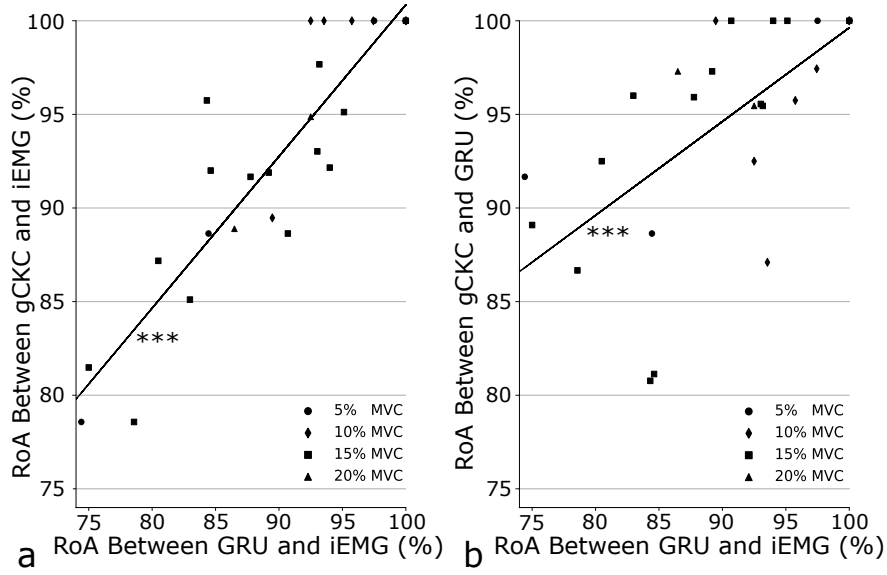


Figure 2.4.2: In **a** the RoA between matched GRU and iEMG sources is plotted against the RoA between the gCKC and iEMG sources for each recording, divided into MVC levels. There is a highly significant positive correlation between the two algorithms. In **b** the RoA between matched GRU and iEMG sources is plotted against the RoA between GRU and gCKC. The smaller correlation implies that the major limiting factor on GRU performance is the quality of the gCKC labels rather than the GRU network model.

A total of 129 MUs were identified across all accepted recordings, of which 31 were matched with sources identified by iEMG decomposition. 22 of the recordings had one match, whilst three had two matches and a single recording three matches. When making a prediction on test data, the calculation of one second (2048×60 samples) of observation matrix took on average 67ms. The median RoA (interquartile range) between gCKC and the GRU network predictions on unseen test data for all sources was 94.5% (80.0% - 100.0%), whilst for the subset of these sources which were matched with iEMG the median RoA between CKC and gCKC was 94.5% (87.5% - 99.1%) and for unmatched sources 94.7% (77.0% - 100.0%). Mann-Whitney U testing showed no significant difference in gCKC vs GRU RoA between matched and unmatched units ($p = 0.465$).

The median RoA between gCKC and iEMG sources (fig. 2.4.1) for the unseen test set was 94.9% (88.8% - 100.0%), whilst the median RoA between GRU-predicted and iEMG sources was 92.5% (84.5% - 97.5%). For individual recordings the gCKC predictions were slightly better than those of the GRU when compared to iEMG, with a median RoA difference of 2.4% (0.0% - 4.4%). There was a significant correlation between the performance of gCKC and GRU as measured by RoA against the iEMG (fig. 2.4.2a), with a Spearman's rho of 0.904 ($p < 0.001$). A smaller correlation of 0.762 ($p < 0.001$) was found for the RoA between GRU and gCKC and the RoA between GRU and iEMG (fig. 2.4.2b). This suggests that the main limiting factor on the performance of the GRU network was the quality of the gCKC decomposition used for training.

When Gaussian noise was added to the test set both gCKC and performed similarly at high SNRs (fig. 2.4.3). As with the original test data, at an SNR of 30 dB the gCKC slightly

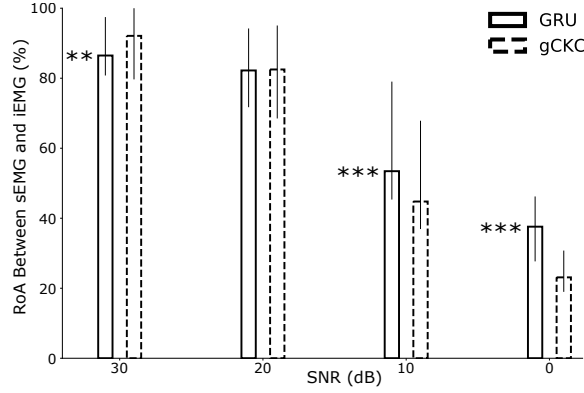


Figure 2.4.3: RoA of GRU and gCKC predicted activations compared to matched iEMG sources for different levels of added noise. Error bars show the interquartile range, with significance calculated by Wilcoxon signed-rank test. The GRU and gCKC perform similarly at low noise levels, but as the SNR falls the GRU begins to significantly outperform gCKC.

outperformed the GRU when compared to matched iEMG sources, with the median RoA difference of 2.1% (0.0% - 6.6%) significant on Wilcoxon signed-rank test ($p = 0.021$). At 20dB there was no significant difference ($p = 0.944$). However, as SNR fell further the GRU predictions began to significantly outperform gCKC, with the GRU RoA improving on the gCKC by a median value of 9.9% (4.5% - 16.0%, $p < 0.001$). This was also true at an SNR of 0dB, with a median difference of 12.8% (5.9% - 17.3%, $p < 0.001$).

2.5 Discussion

The development of algorithms that can accurately decompose sEMG signal into constituent MU activation trains provides invaluable tools for the study of neural control of movement[24]. However, computationally complex preprocessing operations and low robustness to varying levels of signal noise can limit applications. In this study we demonstrate for the first time the direct decomposition of unwhitened sEMG signal into constituent sources, using a supervised deep learning framework. A GRU network trained using output MU activation labels from

the gCKC algorithm quickly decomposed both simulated and experimental unseen sEMG signal into accurate IPTs, with processing latencies of the experimental sEMG signal well below 100ms per second of data.

In simulations the trained GRU network proved extremely robust to the addition of noise in the unseen test set, outperforming the learned separation matrix of gCKC. This behaviour was also seen when noise was added to the experimental data, with the difference increasing as the signal became noisier. Heteroscedasticity is a common issue in surface sEMG as the quality of the contact between electrode and skin can degrade with both movement and perspiration. The GRU also outperformed gCKC when the training data had a low SNR, the implication of which is that the network is not necessarily limited by the performance of the unsupervised algorithm being used to generate training data. A potential explanation is that the network is better able to generalise to new data than the original separation matrix, an effect augmented by the addition of noise during the training process. Also of interest is the relative amplitudes of the spikes in the LMMSE IPT, which can be interpreted as the confidence in an activation being present, weighting the loss propagated through the GRU network during training. This is an important result, as it suggests a role for a deep learning framework in improving the accuracy of the MU activation library used by gCKC to update the separation matrix.

Two-source validation remains the gold-standard method for validating sEMG decomposition algorithms on experimentally-collected sEMG signal[62]. When compared to independently collected and source-separated iEMG the GRU network performed similarly to gCKC on unseen test data. This was despite the network only being trained on 12s of gCKC output, and in the absence of computationally-expensive signal whitening. A limiting factor

on network performance was the accuracy of the underlying gCKC decomposition used to generate the MU activation labels used for training. There was also a weaker correlation between GRU vs gCKC RoA and GRU vs iEMG RoA, suggesting that, unlike the simulation experiments, deviations in GRU predictions from those of gCKC usually caused errors. This is likely a function of the short length of the available experimental sEMG signal.

A further advantage of the proposed methodology is that a GRU network is a fully differentiable function. Within the context of a HMI, a second model is trained to use the decomposed output to classify gestures or joint position[56][57]. Providing the second model is also differentiable, it becomes possible to backpropagate error through the entirety of the conjoined decomposition and classifier model, allowing efficient network updates to be made. In this scenario the decomposition component of the conjoined network is pre-trained with the output of a blind source separation algorithm, meaning that part of the network will already be an efficient feature extractor prior to classifier training. Indeed, the flexibility of the proposed system allows it to adapt as new data is collected, potentially allowing further generalisation to changes in signal such as the activation of new MUs.

The main shortcoming of this study is the short length of the experimental recordings, likely to be the explanatory factor behind the small difference in performance between the experimental and simulated validations. With such a short length even the regularisation and augmentation procedures during training were unable to prevent a small amount of overfitting on the training data, leading to a small gap in performance at high SNRs, before the advantage gained from improved robustness to noise dominates. The low number of matched sources between the sEMG and iEMG signals is also explained by the reduced length of the recordings; despite almost identical decomposition procedures, more sources

were matched in the original study, which used the full 20s for decomposition[62]. Planned future studies will be able to make use of advances in multichannel iEMG to greatly expand the number of matched sources[64].

2.6 Conclusion

The accurate and rapid supervised decomposition of simulated and experimental unwhitened sEMG signals was demonstrated using a gated recurrent unit network trained using the gradient convolution kernel compensation algorithm as a label generator. In relatively noise-free signal the two approaches performed similarly in prediction-accuracy against the ground-truth in simulated data and against simultaneously-recorded iEMG in experimental data, despite GRU network operating directly on unfiltered and unwhitened sEMG signal. Differences between the two methods emerged with low signal-to-noise ratios. In simulated data the GRU network outperformed gCKC on the unseen tests set when the training data was very noisy, despite being trained on the gCKC output. When additional noise was added to the test data the GRU was better able to generalise in both the simulated and experimental datasets. Such a supervised framework offers a potential route to adapting the decomposition algorithm to online HMI applications, as well as potentially being able to improve the accuracy of offline decompositions.

Chapter 3

Pretraining Recurrent Neural Networks with Finite Element Method Simulations of EMG¹

¹This chapter was originally published as *A Myoelectric Digital Twin for Fast and Realistic Modelling in Deep Learning*[65] in *Nature Communications* (2023) with co-authors Kostiantyn Maksymenko², Irene Mendez Guerra³, Samuel Deslauriers-Gauthier⁴ and Dario Farina³. This was a composite work combining an advanced finite element method simulator for surface EMG and a pretraining method for neural networks which utilises the simulation. Whilst not a focus of this thesis, some components of the methodology relating to FEM simulation have been preserved to provide context to those components which relate to the pretraining methodology, for which more details have been added than were in the body of the original paper. The paper has also been changed from an introduction-results-discussion-methods layout to an introduction-methods-results-discussion layout to better suit the thesis, consequently, in aid of logical flow, some of the contents have been moved between sections and others modified. All included images are from the original paper.

² Neurodec, Sophia Antipolis, France

³ Department of Bioengineering, Imperial College London, UK

⁴ Centre Inria d'Université Côte d'Azur, Valbonne, France

Chapter Abstract

Muscle electrophysiology has emerged as a powerful tool to drive human machine interfaces, with many new recent applications outside the traditional clinical domains, such as robotics and virtual reality. However, more sophisticated, functional, and robust decoding algorithms are required to meet the fine control requirements of these applications. Deep learning has shown potential in meeting these demands, but requires a large amount of high-quality annotated data. Data augmentation using simulations, a strategy applied in other deep learning applications, has never been attempted in electromyography due to the absence of computationally efficient models. We introduce the concept of a Myoelectric Digital Twin - highly realistic and fast computational model tailored for the training of deep learning algorithms, demonstrating the benefits of pretraining in the experimental setting on a pipeline of motor unit decomposition from electromyography signal collected from wrist-mounted electrodes.

3.1 Introduction

Biosignals have been classically used for studying the underlying physiology, for clinical diagnostics, and for monitoring. More recently, they have also been used for interfacing humans with external devices. For example, signals measured at the surface of the skin from skeletal muscle electrical activity, i.e. surface electromyography, are used for the control of bionic limbs[66]. In this application, the recorded electrical signals are converted into motion commands using machine learning[67, 68, 69]. In recent years, with the development of deep-learning based methods as well as wearable and cost-effective recording devices, there has been increased interest in using muscular signals as a basis for human-machine interfaces[70, 71]. The potential applications go well beyond the traditional clinical domains of prostheses and orthoses and range from robotic control to gaming and virtual reality[72].

To be useful in real world applications, particularly for the mass market, EMG-based muscle-computer interfaces need to be highly intuitive and functional. They also need to work for the vast majority of subjects with little or no user-specific calibration, regardless of their anatomy and physiology. Traditional machine learning techniques might work well for the relatively simple task of hand gesture classification, but they require training on a specific subject and potential regular recalibrations[66, 72]. This is a major bottleneck for mass market applications, where obtaining subject specific data is not possible. Deep learning methods could tackle these problems[73], but large, annotated, and high-quality datasets are required. To consider deep learning, training data needs to be recorded for different subjects, at different times, with high variability in electrode configurations and experimental paradigms. In addition, it is challenging and in some cases impossible to properly describe the underlying physiological or neural parameters (e.g. individual muscle forces, fibre physiological param-

eters, motor neuron impulse timings), which are crucial for the correct annotation of data samples. As a result, acquiring experimental EMG data in sufficient quantity and quality is not only expensive and time-consuming, but in many cases not possible.

Data augmentation via simulation is an alternative approach to lengthy data acquisitions, and indeed, augmentation techniques have been recently introduced for electrophysiological signals [74, 75, 76, 77]. However, most of these augmentation methods use black-box models, which aim to capture essential features of the signal without relating them to the underlying physiology[78]. Thus, the ground truth for most of the crucial parameters is still unknown, greatly limiting the potential use cases of such approaches. More sophisticated biophysical modelling methods are based on solving so-called forward equations (e.g., Poisson equation in the electrostatics case). However, this type of biophysical modelling has not been considered in the context of data augmentation for machine learning approaches. Indeed, state-of-the-art models are either not sufficiently realistic or not computationally efficient to produce suitable training data. For example, in the case of describing the generation of EMG signals, analytical models based on simple geometries of the tissues[79, 80, 81, 82, 83] provide simulations that reflect the broad characteristics of the signals, but cannot be used to reproduce specific experimental conditions due to the overly simplified anatomy. The more realistic models of EMG generation based on numerical solutions of the Poisson equation with generic volume conductor shapes[84, 85] are currently limited by their prohibitive computational time.

Here, we describe an EMG simulation method (Fig. 3.1.1), based on the numerical solution of the forward equations suitable for deep learning data augmentation. It produces highly realistic EMG recordings, provides access to all underlying physiological parameters, and is significantly more computationally efficient, compared to the state-of-the-art methods.

Although, there is no benchmark to evaluate and compare the exact performance of different simulation methods, our model takes minutes (or even seconds) to perform simulations that take hours with the state-of-the-art methods [86, 85]. We achieved this gain in computational efficiency not by simply optimizing the code or using parallel computing, but by exploiting the mathematical and structural properties of the model and reformulating them theoretically. Our results show that it is possible to simulate EMG signals for anatomically accurate conductor geometries and multiple muscles with tens of thousands of muscle fibres in a few seconds.

Furthermore, we show a potential use of high volumes of simulated surface EMG data for deep learning, utilising the proposed model to generate data which can be used to pretrain neural networks. This methodology is used in other deep learning domains, such as the use of the ImageNet image database to pretrain object classifiers prior to adaptation to specific applications[87]. The selected application was that of a time series classifier, which took as an input unwhitened high-density surface sEMG signals recorded from wrist-mounted electrodes and outputted discrete motor unit activation events. We show that pretraining with the simulation outputs greatly improves the efficiency of the model training pipeline when compared to random initialisation.

3.2 Methods

3.2.1 Forward Problem

The fibre extracellular potentials that are measured by EMG electrodes are generated by transmembrane currents. The properties of bioelectric currents and potential fields can be

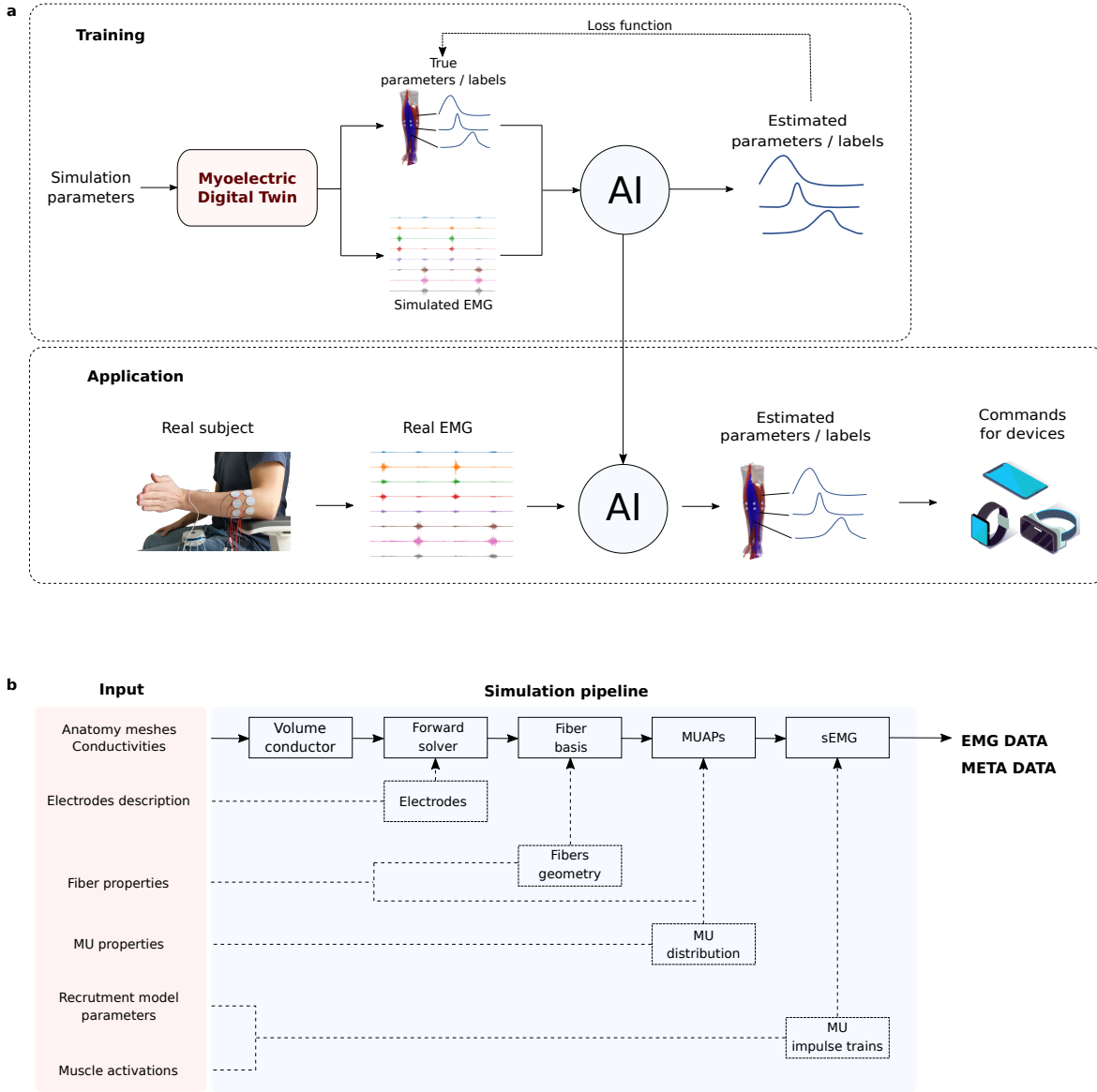


Figure 3.1.1: **a** General strategy of using the Myoelectric Digital Twin to train an artificial intelligence, that then can be used to process real surface electromyography signals in real world applications. **b** Schematic representation of the software's input, main elements of the simulation pipeline and output. User can define a large set of simulation parameters describing anatomical and physiological properties of the tissues, geometry and montage of electrodes, fibres geometry and their physiological properties, motor unit distribution and their recruitment model, activation of individual muscles, etc. The main components of the simulation pipeline are described in details in Methods section. For a given set of the parameters, the software outputs the resulting simulated EMG data as well as all the metadata of the simulation (e.g. individual motor unit action potentials).

determined from solutions of the Maxwell's equations, taking into account the electrical properties of biological tissues. Because of the relatively low frequencies of signal sources of biological origin, the quasi-static assumption can be applied[88, 89], so that the electric potential and the primary current sources are related by the following Poisson equation[88, 90, 59] with Neumann boundary conditions:

$$\begin{cases} \nabla \cdot (\sigma \nabla \phi) = -I & \text{in } \Omega \\ \sigma \frac{\partial \phi}{\partial \mathbf{n}} = \sigma \nabla \phi \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \end{cases} \quad (3.2.1)$$

where $\Omega \subset \mathbf{R}^3$ is a volume conductor domain of interest, $\partial\Omega$ its boundary with outward pointing normal unit vector $\mathbf{n}$, $\phi(\mathbf{r})$ [V] is the electric potential, $I(\mathbf{r})$ [A/m³] is the current source density (CSD), $\sigma(\mathbf{r})$ [S/m] is a conductivity tensor.

The second line of the equation (boundary condition) reflects the assumption that no current flows out of the domain of interest. In the context of EMG modelling, this implies that there is no current flow between the skin and air. The current source density $I(\mathbf{r})$ is interpreted as the volume density of current entering or leaving the extracellular medium at position $\mathbf{r} \in \Omega$. A negative CSD corresponds to current leaving the extracellular medium (due to the fibre transmembrane currents) and is thus conventionally called a sink. Likewise, current entering the extracellular medium is called a source[91, 92].

Equation (3.2.1) cannot be solved analytically for general volume conductor geometries, but several numerical methods can be used to approximate its solution. Here, we use the finite element method (FEM)[93], which discretizes the volume conductor Ω as a tetrahedral mesh Ω_t . Given this mesh, we use the Galerkin method to project the potential ϕ onto the space of piecewise affine functions defined on Ω_t . Fig. 3.2.1a and Fig. 3.2.1b illustrate an example

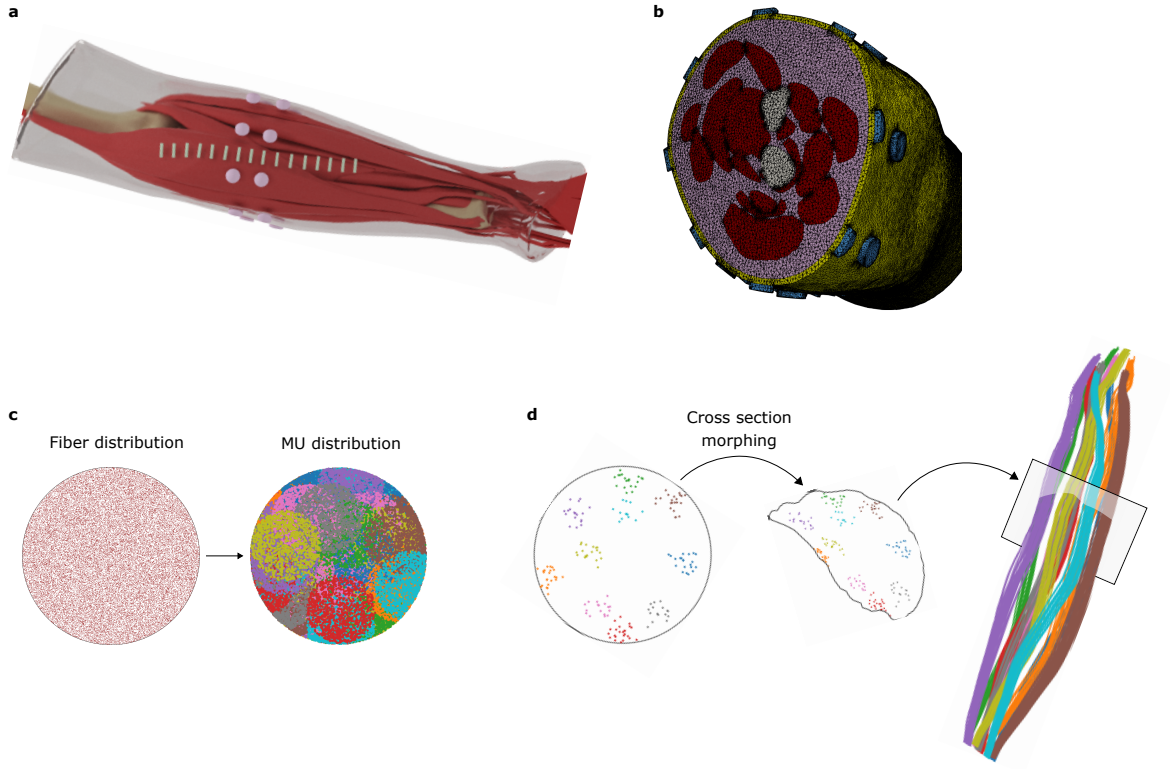


Figure 3.2.1: **a** Surface geometry of muscles, bones, subcutaneous tissue, skin and electrodes used for arm modelling (taken from BodyParts3D, The Database Center for Life Science). **b** Cross-section of the volume mesh generated from the arm surfaces. **c** Uniformly distributed fibres inside a unit circle are grouped into motor units (MUs) of different sizes, locations and territories. **d** Example of mapping of 10 small motor units from the circle into an arbitrary muscle by morphing the unit circle into the muscle cross section.

of a realistic forearm model and corresponding discretized volume mesh, respectively.

This discretization process converts the continuous operator problem of equation (3.2.1) to a finite system of linear equations:

$$A\mathbf{v} = \mathbf{b} \quad (3.2.2)$$

where A is a symmetric and sparse $n_v \times n_v$ matrix, n_v is the number of mesh vertices, $\mathbf{v} \in \mathbf{R}^{n_v}$ is a vector of potential values at mesh nodes, and $\mathbf{b} \in \mathbf{R}^{n_v}$ is a vector containing source information. Because the electric potential is defined up to a constant, the matrix

A always has a one dimensional null space. To obtain a unique solution to the system of equation (3.2.2), we constrain potentials $\mathbf{v}$ to have a zero sum.

In the context of EMG, we are not interested in finding electric potentials everywhere in the conductor, but only at the electrode locations. Let S be a selection matrix with a shape $n_e \times n_v$ which only selects the values at EMG electrode locations (n_e is the number of electrodes). Each row of S can be designed to select a single point location or to integrate over an area (e.g. the electrode-skin interface) depending on the location and number of its non-zero elements. Also, let $\mathbf{b}(\mathbf{r})$ correspond to a point source at location $\mathbf{r}$. The resulting EMG signal is thus given by:

$$\mathbf{v}_{\text{point}}(\mathbf{r}) = SA^{-1}\mathbf{b}(\mathbf{r}). \quad (3.2.3)$$

Let us analyse in more detail the structure of A and $\mathbf{b}$ from equation (3.2.2). Let $\{w^i(\mathbf{r}), i = 1 \dots n_v\}$ be a set of n_v P^1 (piecewise linear) basis functions over the tetrahedral mesh Ω_t . Note, that w^i is 1 at the i -th vertex of the mesh, is 0 at all other vertices and is linear at all tetrahedra adjacent to the i -th vertex. In this case, A and $\mathbf{b}$ have the following structure:

$$A_{ij} = \int_{\Omega_t} \sigma(\mathbf{r}) \nabla w^i(\mathbf{r}) \nabla w^j(\mathbf{r}) d\mathbf{r}$$

$$b_i = \int_{\Omega_t} I(\mathbf{r}) w^i(\mathbf{r}) d\mathbf{r}.$$

First, let us notice that A is symmetric and, in general, a very large matrix which can be stored only because it is sparse. Indeed, the functions w^i have a compact support and their

Chapter 3

pairwise scalar product is non-zero only for neighbour functions. Since the pseudo-inverse (or the inverse) of a sparse matrix is usually not a sparse matrix, it is impractical to compute it because of the amount of memory needed to store it. Thus, iterative methods are typically used to solve the system of equation (3.2.2) for every given $\mathbf{b}$.

Consider the case of $I(\mathbf{r}) = \delta(\mathbf{r} - \bar{\mathbf{r}})$ which corresponds to a unit point current source at a location $\bar{\mathbf{r}}$. Without loss of generality, we assume that this source is inside a tetrahedron formed by the vertices $i_1, \dots, i_4$ of the mesh. In this case, we obtain:

$$b_i = \begin{cases} \lambda_j, & \text{if } i \in \{i_1, \dots, i_4\}, \\ 0, & \text{otherwise} \end{cases}$$

where $\{\lambda_j, j = 1, \dots, 4\}$ are the barycentric coordinates of the point $\bar{\mathbf{r}}$ inside the tetrahedron $\{i_1, \dots, i_4\}$. Applying this expression to equation (3.2.3), we obtain:

$$\mathbf{v}_{\text{point}}(\bar{\mathbf{r}}) = SA^{-1}\mathbf{b}(\bar{\mathbf{r}}) = SA^{-1}\bar{B}\lambda.$$

where $\bar{B}$ is a $n_v \times 4$ matrix with $\bar{B}_{i_j, j} = 1$ for $j = 1, \dots, 4$, and 0 otherwise. This implies that the solution of the system of equation (3.2.2) for any unit point source can be computed as a barycentric sum of solutions on the vertices of the corresponding tetrahedron. Therefore, it is sufficient to compute solutions of equation (3.2.2) for basis sources located on mesh vertices, to be able to evaluate a solution for any point inside this mesh efficiently. Let n_s be the number of such basis sources. For the most general case, when the source can be located anywhere inside the mesh and $n_s = n_v$, let B be a $n_v \times n_s$ identity matrix. The objective is to compute basis solutions:

$$V_{\text{basis}} = SA^{-1}B \quad (3.2.4)$$

where V_{basis} is a $n_e \times n_s$ matrix, whose columns contain the solutions of equation (3.2.2) for a unit point source located at the corresponding mesh vertex. Hence, the potentials for any source location $\mathbf{r}$ is given by:

$$\mathbf{v}_{\text{point}}(\mathbf{r}) = V_{\text{basis}}\lambda(\mathbf{r}) \quad (3.2.5)$$

where $\lambda(\mathbf{r}) \in \mathbf{R}^{n_s}$ is a vector, whose four non-zero elements contain the barycentric coordinates of a point $\mathbf{r}$ inside a corresponding tetrahedron. Note, that one may restrict potential sources to be located inside specific subdomains of the whole mesh (which is the case for EMG). In this case, n_s corresponds to the number of vertices of these subdomains, and the matrix B is a submatrix of the identity matrix.

The most straightforward way to compute V_{basis} from equation (3.2.4) is to solve a problem of the form $A\mathbf{x} = \mathbf{b}_i$ for each column of the matrix B . It would thus require solving n_s systems of linear equations. For realistic conductor geometries, which have a large number of vertices, solving a single system may take up to a few minutes and solving n_s systems quickly becomes impractical. Therefore, we propose the use of the adjoint method[94], which requires solving n_e systems only. In the context of EMG, the number of electrodes is usually significantly smaller than the number of vertices in the muscle subdomain meshes, i.e. $n_e \ll n_s$. Let us define $K = SA^{-1}$, which is a matrix of size $n_e \times n_v$. Because A is symmetric, and the inverse of a symmetric matrix is also symmetric, we can write $K^T = A^{-1}S^T$. Then, K can be found by solving the system:

$$AK^T = S^T. \quad (3.2.6)$$

The matrix S^T has n_e columns and, thus, only n_e linear systems need to be solved to find K . The basis solutions can then be found as:

$$V_{\text{basis}} = KB. \quad (3.2.7)$$

3.2.2 EMG Signal of a Single Fiber

The action potential generated by the flow of ionic currents across the muscle fibre membrane is the source of excitation. For a given intracellular action potential (IAP) model $V_m(z)$, the transmembrane current source per unit length is proportional to the second derivative of $V_m(z)$, where z is a fibre arc length measured in mm . A general description of the current source density travelling at velocity v along the fibre with the origin at the neuromuscular junction at location z_0 is [95, 96, 97]:

$$I(z, t) = \sigma_{\text{in}} \pi r^2 \cdot \frac{\partial}{\partial z} \left[\psi(z - z_0 - vt) w_{L_1}(z - z_0 - \frac{L_1}{2}) - \psi(-z + z_0 - vt) w_{L_2}(z - z_0 + \frac{L_2}{2}) \right] \quad (3.2.8)$$

where $z \in [0, L]$ is a location along the fibre of length L , $\psi(z) = \frac{d}{dz} V_m(-z)$, L_1 and L_2 are the semi-lengths of the fibre from the end-plate to the right and to the left tendon, respectively, σ_{in} is the intracellular conductivity, and r is the fibre radius. We have chosen w_L to be a Tukey window, as proposed in [86]. The intracellular action potentials (IAP) V_m [$\frac{mV}{mm}$] can be

mathematically described in the space domain as proposed in[98]:

$$V_m(z) = 96z^3e^{-z} - 90.$$

Let $\mathbf{r}(z)$ be a fibre geometry parametrized with respect to the fibre arc length z . Combining the transfer function of a point source in equation (3.2.3) with the fibre's current density in equation (3.2.8), we obtain the equation for the EMG signal resulting from a single fibre activation:

$$\mathbf{v}_{\text{fibre}}(t) = \int \mathbf{v}_{\text{point}}(\mathbf{r}(z))I(z, t)dz. \quad (3.2.9)$$

This integral can be efficiently approximated by discretizing the fibre geometry into sufficiently dense spatial samples $\{\mathbf{r}(z_i)\}_i$ and assuming that $\mathbf{v}_{\text{point}}(\mathbf{r}(z))$ is piecewise constant around these points. If we also rewrite equation (3.2.8) in a shorter form as $I(z, t) = \sigma_{\text{in}}\pi r^2 \cdot \frac{\partial}{\partial z}F(z, t)$, equation (3.2.9) becomes:

$$\begin{aligned} \mathbf{v}_{\text{fibre}}(t) &\approx \sum_i \mathbf{v}_{\text{point}}(\mathbf{r}(z_i)) \int_{z_i - \Delta_i}^{z_i + \Delta_i} I(z, t)dz = \\ &\sum_i \mathbf{v}_{\text{point}}(\mathbf{r}(z_i)) \int_{z_i - \Delta_i}^{z_i + \Delta_i} \sigma_{\text{in}}\pi r^2 \cdot \frac{\partial}{\partial z}F(z, t)dz = \\ &\sigma_{\text{in}}\pi r^2 \sum_i \mathbf{v}_{\text{point}}(\mathbf{r}(z_i)) \left(F(z_i + \Delta_i, t) - F(z_i - \Delta_i, t) \right). \end{aligned} \quad (3.2.10)$$

Note, that $\mathbf{v}_{\text{point}}(\mathbf{r}(z_i))$ can be efficiently computed from equation (3.2.5). Moreover, once $\mathbf{v}_{\text{point}}(\mathbf{r}(z_i))$ are computed for all given fibres, we can change the parameters of the current source density (action potential waveform shape, propagation velocity, location of neuromus-

cular junction), and compute the corresponding EMG signal with equation (3.2.10) by only matrix multiplication complexity.

3.2.3 Modelling of Motor Units

The motor unit action potential is the summation of the single fibre action potentials of the muscle fibres in the MU. Different types of MUs can be modelled[99, 100]. Our approach consists in generating fibre and motor unit distributions inside a unit circle, and then projecting it into arbitrary 3D muscle geometry (Fig. 3.2.1d), using methods similar to those described in[101]. This provides a high level of control for the fibre and MU distribution parameters independently of a particular muscle geometry. A common way to simulate fibres and MUs is to start by defining MU positions, sizes and territories, and then simulate fibres inside these MUs according to their parameters[60, 102]. We, however, propose another approach. First, we simulate uniformly distributed fibres inside a unit circle. Then, MU centres and their circular territories are generated and, finally, we associate each fibre to an MU. A fibre is associated to one of the MUs that contains it inside its territory with a probability proportional to the MU density (Fig. 3.2.1c). This approach has two main advantages. First, it guaranties (by construction) the uniform fibre distribution inside a circular muscle cross-section. Second, once fibres are generated and projected into a muscle geometry, different MU distributions can be generated very quickly, without regenerating fibres and recomputing transfer functions $\mathbf{v}_{\text{point}}(\mathbf{r}(z_i))$ for their nodes.

3.2.4 Pretraining a Neural Network

We examined whether a neural network that was pretrained using simulated data and then fine-tuned on individual experimental sEMG signal outperformed a neural network that used only a random initialisation, selecting as a task the decomposition of sEMG into its constituent MU activity. Specifically, the myoelectric digital twin simulation (Fig. 3.3.2b) was used to generate a large library of MUAP templates, which were then used to pretrain a neural network architecture using a multitask methodology. A component of the pretrained network was then used as the initialisation for the training of a decomposition network on experimental sEMG.

The network was pretrained using simulation-generated MUAP templates. Four digital twins were created using different model parameters (tissue conductivities, MU distribution, fibre properties, etc.), with 320 MUAP templates generated from simulations of flexor digitorum profundus and superficialis. Training a classification network with an output dimension of 320 is very unstable when the target data is highly sparse and class-imbalanced, so instead a multitask framework was used consisting of a multi-headed network with shared body network. The overall task of training a classifier on 320 MUAPs was subdivided into 64 individual tasks, with each head operating on a single task of predicting a set of five MUAPs from sEMG. The parameters of the body network were used and updated simultaneously for all 64 tasks, i.e. a hard parameter sharing paradigm[103]. In this way, the body network could be pretrained to find an efficient featurisation, which could then be used as a starting point for fine-tuning on the final prediction task with experimental wrist sEMG signals.

Each set of five MUAPs was used to generate windows of signal with a range of MUAP superpositions (Fig. 3.2.2a). In signal windows with motor neuron activity a MUAP template

was placed in the centre of the window, before being additively superimposed with a random number of MUAP templates from other motor units at random time offsets. In windows without activity no template was placed in the centre of the window. In this way the body layer was trained to act as a more general feature extractor, whilst the individual heads made class predictions specific to each recording.

To use the simulation-pretrained network on the experimental data the parameters from the body network were used (Fig. 3.2.2b), whilst the head network used a random initialisation (Kaiming uniform). This was compared to a random initialisation of both the body (uniform) and head network (Kaiming uniform). The network was optimised identically for both types of initialisation, i.e. the only difference was whether the body network was simulation-pretrained or not. This included fixing the architecture, no further hyperparameter optimisation was conducted. To compare the two resultant neural networks for sEMG decomposition, predicted activity on the held-out test data was first converted to spike timestamps using a two-class K-means clustering algorithm and then compared to the ground truth using the rate of agreement metric (RoA).

3.2.5 Neural Network Architecture and Training

A gated recurrent unit (GRU) neural network was used as the body of the deep learning model due to previous studies showing good performance with this data type[43]. The architecture of the GRU network was selected through grid search hyperparameter optimisation, and consisted of a single GRU cell with a hidden dimension of 1024[104]. This was used to convert 130-channel sEMG to a 1024-channel featurised signal, of which 20 samples were taken into a task head, consisting of a densely-connected linear layer with sigmoid activation

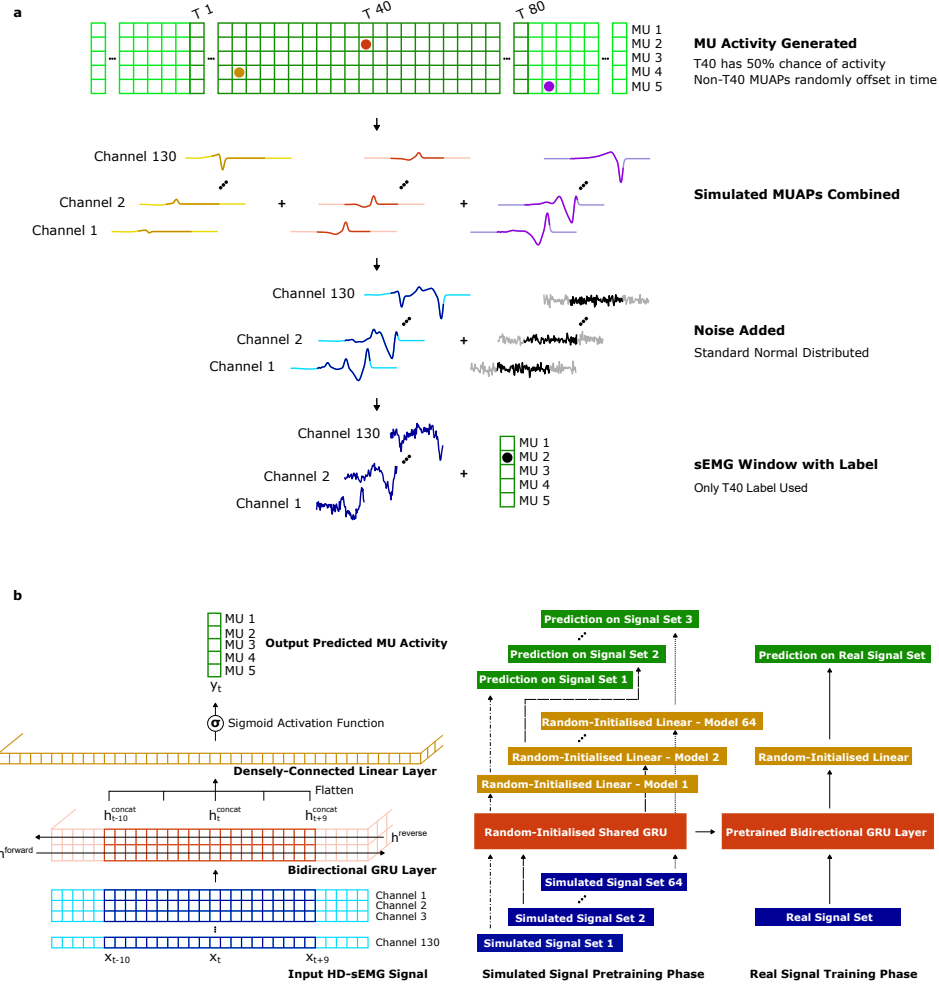


Figure 3.2.2: **a** Methodology used to build windows from the simulated MUAP template set for the pretraining phase. Each simulated template was 160 samples wide at a 2048Hz sampling rate and with 130 channels. First, either a MUAP template was placed in the centre of the window or it was left empty at a 50% probability. Then MUAP templates from other MU classes were added to the window at a random offset to generate superpositions. Finally, standard normal distributed noise was added to the window, with the central 80 samples then paired with the label for supervised learning. **b** The neural network architecture and pretraining methodology used to improve the performance of a deep learning-based sEMG decomposition algorithm. The neural network consists of a single gated recurrent unit layer, with predictions made using a 20-sample wide window of the hidden vector output, which is flattened before being passed to a sigmoid-activated densely-connected linear layer. In the pretraining phase a multitask learning regimen is used to optimise the parameters of the gated recurrent unit using the simulated sEMG. This pretrained layer can then be used to improve the optimisation performance on real sEMG data.

function, for final prediction of MU activity for each time step.

Optimisation workflows were kept constant between both pretraining on the digital twin data and training on the final task of EMG decomposition. Binary cross entropy was used as the loss function and Adam with weight decay used as the optimising algorithm[40]. To improve model generalisation an early-stopping framework was used, based on 10% of the training data retained as a validation set. Training, validation and test data was z-score standardised using the mean and standard deviation calculated from the training set. During optimisation the input signal was augmented with noise of standard normal distribution. To account for the high sparsity of the output matrix, samples containing motor neurons were artificially oversampled, with each input batch of 512 time instants containing at least 20% motor neuron activation. All machine learning was implemented using the Pytorch library in Python.

3.2.6 Experimental Validation

To evaluate the effect of using the simulation-pretrained network, an experimentally collected high-density sEMG signal dataset was used, originally created to test low-force human-computer interaction with wrist-wearable interfaces [72]. The experimental protocol was designed in agreement with the Declaration of Helsinki and was approved by Imperial College London ethics committee (JRCO: 18IC4685). A balanced gender distribution (as self-reported) was prioritised during participants' recruitment. Nine participants (4 females, 5 males, ages: 23-31) took part in the study after signing informed consent forms. Participants were not compensated for their involvement in the experiments. The participants performed 5-second isometric contractions of their dominant-hand index finger at 15% of

maximal force, with sEMG activity measured using two flexible 5x13 electrode grids with 8-mm spacing placed on the circumference of the wrist, immediately proximal to the ulnar head. Contraction forces were measured by a force cell and fed back to the user on a computer monitor.

sEMG signal was acquired using a Quattrocento amplifier (OT Bioelettronica) at 2048Hz sampling frequency, whilst force profiles were sampled with a Phidget load cell at 10Hz. A Matlab 2019b (The MathWorks, Inc) program was used to synchronise both modalities. The sEMG signal was then decomposed into motor neuron activity using convolutive blind source separation [14]. For the purpose of training and testing the supervised decomposition pipeline, motor neuron activity was accepted if it was present for at least 80% of the contraction window. For each participant the sEMG signal and accompanying decomposed motor neuron activity (as a sparse binary matrix) was then split into a 4 second training window and a 1 second testing window. This process was conducted once per participant, with all plotted variance being across participants.

3.3 Results

3.3.1 Simulation Outputs

To evaluate the performance of the simulator at multiple scales, we started by simulating EMG signals associated to a single fibre activation inside the brachioradialis muscle. The signal recorded by an array of 16 rectangular electrodes (15 differential channels) when a single fibre was active is shown in Fig. 3.3.1a. The volume conductor model is based on an anatomically accurate forearm geometry, which includes all the muscles, bones, fat, and skin

tissues.

Different distinctive features are present in the simulated signal that are also observed in experimental EMG signals[17]. In particular, electrodes of channel 4 are located on different sides of the neuromuscular junction and thus the respective signals cancel each other out. Channels 7-11 present propagating EMG components resulting from the fibre action potentials propagating from the NMJ to the tendons. Channels 2-6, as well as channels 12-15, contain non-propagating sEMG components, which are due to the AP generation at the NMJ and its extinction at the tendon (end-of-fibre effect), respectively.

A further example is a simulation of an excitation of a single muscle, illustrated in Fig. 3.3.1b. A simple excitation drive for the Brachioradialis muscle is simulated as gradually increasing from 0% to 100% of the maximum voluntary contraction and smoothly decreasing back to 0%. For the purpose of demonstration, we simulated a 1-min-long, 100% maximum voluntary contraction excitation of the Brachioradialis muscle with 50000 individual fibres and 200 motor units. The mesh of the volume conductor contained 2.1M vertices, which formed 13M tetrahedra. 16 rectangular and 16 circular electrodes were included in the model. The signal was simulated for 8 circular bipolar electrodes located around the forearm. In this example, the volume conductor effect becomes particularly visible with electrodes nearer to the active muscle having higher signal amplitudes. Notice that the electrodes record different signal waveforms as the muscle units are located at varying distances from the electrodes, weighting their contribution to the observed EMG signals. We also observe an increase of the signal amplitude with muscle excitation, an important feature of experimental EMG signals, which is a consequence of progressive motor unit recruitment and of an increase in the discharge rates of the active motor units.

Finally, we simulated sEMG signals from multiple muscle excitations, corresponding to the active wrist flexion and extension and passive wrist abduction against gravity. We used a simple muscle excitation model for three groups of muscles (flexors, extensors and abductors). Fig. 3.3.1c and Fig. 3.3.1d clearly show the qualitative similarities in signal characteristics between experimental and simulated data. Our model was able to reproduce the different signal patterns during both flexion and extension. Beside the different activation across the electrodes during flexion and extension, the effect of wrist abduction is also visible in both data sets. Thus, channels 2, 3 and 7, 8 present a small signal activity during the whole duration of the simulation, and not only during flexion/extension peaks. Similar activity can also be seen in experimental data, with channels 2 and 7 being the most active.

3.3.2 Pretraining Results

The simulation pretrained network outperformed random initialisation in decomposition accuracy when compared to the original decomposition as measured by the rate of agreement metric calculated on a one second segment of data[95] (Fig. 3.3.2c). The RoA metric efficiently incorporates false positive and false negative into a single metric of accuracy relevant to the decomposition of discrete events in time series data. The median (IQR) RoA of the pretrained network was 93.8% (84.8 to 100.0), compared to 82.4% (71.6 to 100.0) in the random initialisation network, with a median difference of 8.1 (Hodges-Lehmann estimator, 95% CI 3.4 to 13.3 using method of Walsh averages), significant according to the two-tailed Wilcoxon signed-rank test ($p = 0.00006$, $Z = 4.0$). Of the 39 decoded motor units, 22 had improved RoAs with pretraining and one had a worse RoA, with the remaining 16 showing no change, generally because the initial RoA was already 100% without pretraining (these

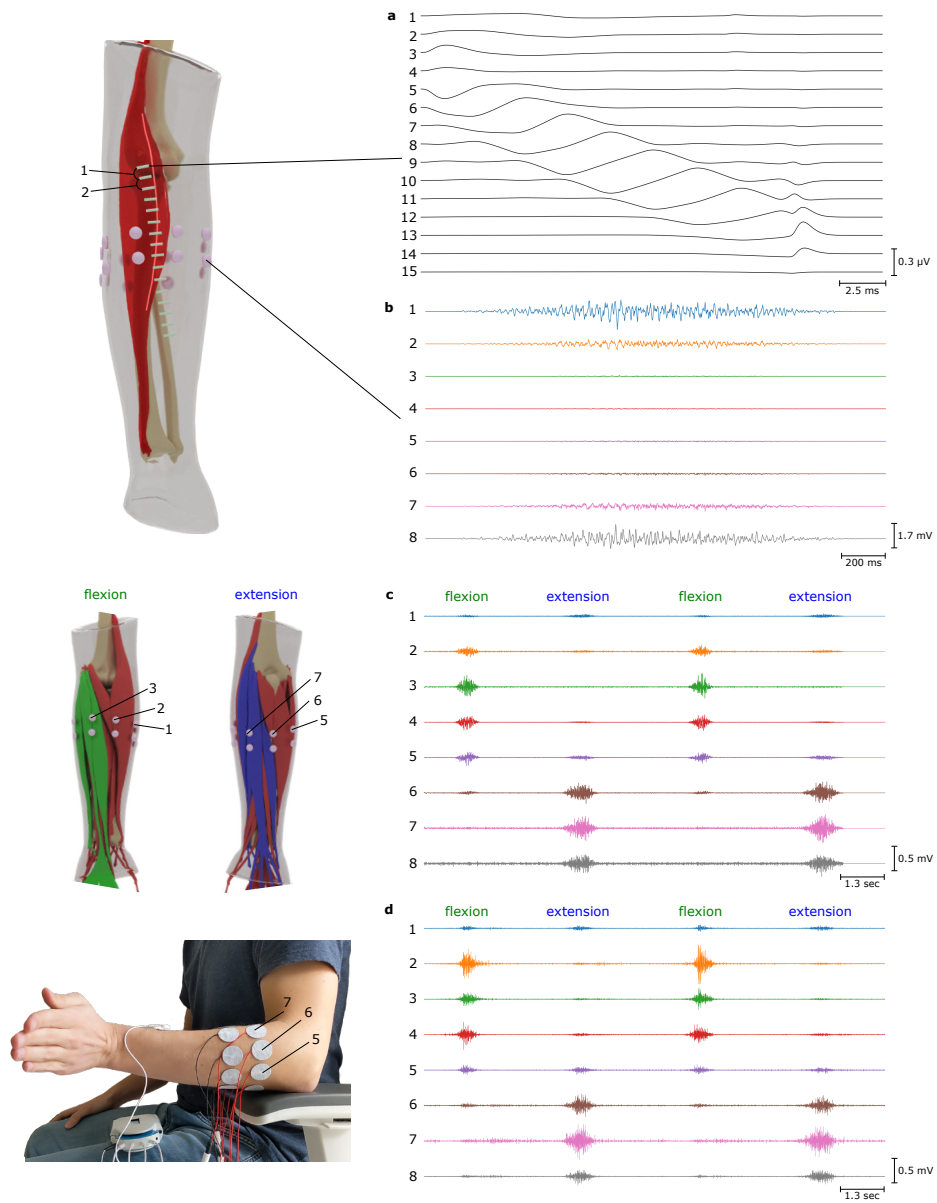


Figure 3.3.1: **a** Single fibre activation in the Brachioradialis muscle measured by an electrode array with 15 differential channels. **b** 2-seconds long activation of the Brachioradialis muscle, reaching 100% of maximum voluntary contraction. 8 bipolar electrodes located around the forearm are simulated. **c** Simulation of wrist flexion and extension by activating the corresponding flexor and extensor muscles. **d** The experimental EMG signals of wrist flexion/extension.

are large-amplitude MUAPs that are simple to detect). The pretrained network had a much lower variance in the accuracy of predictions on the test sets than random initialisation, quickly optimising to a model effective for generalisation to new signals.

When units were subdivided into those from female (18 units) and male (21 units) subjects, both subsamples also showed a significant difference. For female subjects the median difference between trained and random initialisation was 9.1 (Hodges-Lehmann estimator, 95% CI 0.0 to 20.3 using method of Walsh averages) and significant (two-tailed Wilcoxon signed-rank, $p = 0.00694$, $Z = 2.7$). For male subjects the median difference between trained and random initialisation was 5.7 (Hodges-Lehmann estimator, 95% CI 1.6 to 9.7 using method of Walsh averages) and significant (two-tailed Wilcoxon signed-rank, $p = 0.00064$, $Z = 3.4$).

3.4 Discussion

We have proposed an efficient computational approach to highly realistic surface EMG modelling. The method provides the solution to the generation of EMG signals from anatomically accurate volume conductor properties and number of muscle fibres, within limited computational time compatible with real-time signal generation. The proposed model is the only available EMG simulator with realistic description of the volume conductor and optimized for such computational efficiency. The main value of the model is that it opens perspectives for using simulated sEMG for data augmentation in the deep learning framework, something that was never done before using state-of-the-art simulation methods.

The computational efficiency in the volume conductor solution has been recognized as an important component of EMG modelling, and some attempts to decrease the computational

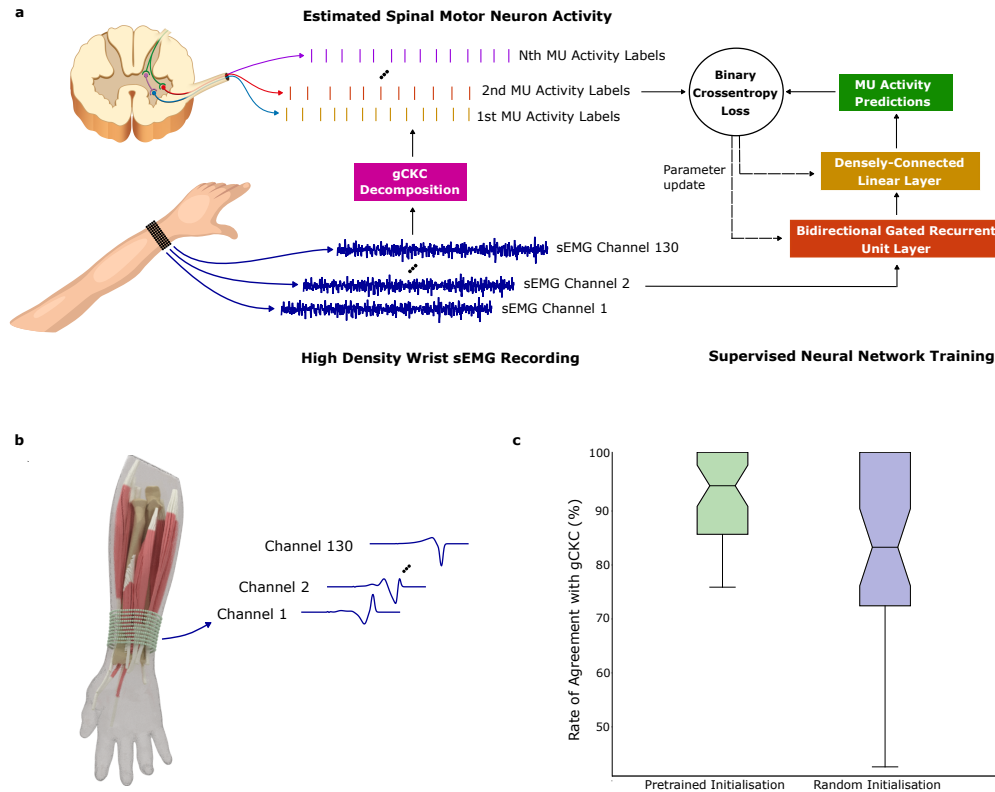


Figure 3.3.2: **a** Decomposition of experimental high density surface electromyography recordings into underlying spinal motor neuron activities. The results obtained with the neural network were tested against the decomposition by a reference blind source separation method and manual editing by an expert operator. MU stands for motor unit. **b** Myoelectric digital twins were used to generate motor unit action potential templates for different muscles and different model parameters (tissue conductivities, fibre properties, tendon sizes, etc). 64 sets, each containing 5 simulated MUAPs for 130 electrodes, were used for pretraining. **c** Rate of agreement (%) between the neural network MU activity predictions and the decomposition algorithm on one second of wrist flexor sEMG signal. Data is presented as median values over 39 motor units from nine participants $\pm$ the interquartile range on the bounds of the box, with the lowest and highest values as the whiskers. Both outputs were converted to timestamps using a two class K-means clustering. The neural network using a gated recurrent unit network that was pretrained using simulated EMG signal significantly outperformed a GRU with random initialisation (Two-tailed Wilcoxon signed-rank test, $Z = 4.0$, $p = 0.00006$, median difference 8.1 using Hodges-Lehmann estimator, 95% CI 3.4 to 13.3 using method of Walsh averages).

time in EMG simulations have been described. For example, the approaches developed by Dimitrov & Dimitrova [105] and Farina et al.[96, 106] substantially decreased the computational time in analytical EMG modelling for simple volume conductor geometries. These models provide simulations which reflect the broad characteristics of EMG signals, but can not be anatomically accurate because of the restrictions on the volume conductor and fibre source geometry. Realistic models using numerical solutions have also been recently proposed. The previous most complete and efficient model has been proposed by Pereira Botelho et al.[85]. These authors have used an anatomically accurate model to simulate EMG signals generated during index finger flexion and abduction. They gained computational speed by using the principle of reciprocity. In fact, one part of our calculations also includes the adjoint method, which is an algebraic representation of this principle. By using reciprocity, Pereira Botelho et al.[85] reported a computational time of 1 hour for simulating the activation of nearly 15500 fibres for 5 electrodes. This time, however, remains impractical for simulating arbitrary large data sets for a variety of parameter values. The model we proposed in this paper significantly surpasses the computational efficiency reported in [85]. We achieved it by efficiently exploiting mathematical properties of the forward equations, in particular by introducing the concept of basis points and by separating model parameters and variables into independent computational blocks. Our approach does not only reduce the computational time for a full simulation, but also allows us to scale the solution, so that new solutions for the same volume conductor can be obtained without re-computing the volume conductor transformation. In this way, the generation of EMG signals within the same volume conductor, but varying all other simulation parameters, can be performed in even shorter time. Complex EMG signals from tens of thousands of muscle fibres located in multiple muscles, can be generated (and

regenerated with different parameter values) in a computational time of the order of seconds.

Some limitations remain in the current state of the presented model. It does not include some sources of variability that are present in experimental EMG signals and strongly impact their processing and analysis. For example, the model does not include advanced noise and artefacts descriptions, biomechanical modelling of the musculoskeletal system, and non-stationary volume conductor properties and fibre geometry.

Also, the automated modelling of muscle forces (or % MVC) for specific movements is out of the scope of this work. We recognize, however, the value of such a tool, and we see two potential ways to approach this task. First, by integrating our myoelectrical model with biomechanical modelling of the musculoskeletal system. This would allow users to define a movement by the dynamics of degrees of freedom (e.g. of a hand). Then, inverse dynamics could be used to estimate the corresponding muscle forces and % MVC, that can be used as an input to our current model. Another possibility is to estimate individual muscle activity from real EMG data. This would require an inverse problem solving algorithm. As a matter of fact, our Myoelectric Digital Twin is, practically, the only way to obtain labelled data for training such inverse algorithms, because the exact muscle forces of real subjects are not, in general, accessible. While these aspects are beyond the scope of this paper, they are relevant features to include in future developments.

The advances presented in this work, together with the proposed future developments, naturally lead to the concept of a myoelectric digital twin - a hyperrealistic, personalized, computationally-efficient model which generates EMG data in a quality and quantity sufficient not only to augment but to replace real data, with utility for AI training in the various real world applications. Here, we have illustrated the potential of this approach by augment-

ing training data for deep neural networks, with the aim of identifying the discharge times of spinal motor neurons from surface EMG signal. By using the simulator to augment training (through a pretraining procedure), we showed an increase in the performance of the decomposition network when applied to experimental data, demonstrating a highly relevant use of the proposed approach for decreasing the need for experimental training data in human-machine interfacing applications.

Chapter 4

Using Deep Metric Learning to Build Self-Correcting MU Decomposition Pipelines¹

¹This chapter was originally published as *Deep Metric Learning With Locality Sensitive Mining for Self-Correcting Source Separation of Neural Spiking Signals*[107] in *IEEE Transactions on Cybernetics* (2023) with co-author Dario Farina². As with the previous chapters, some alterations have been made to better suit the overall thesis structure and duplication removed from the theory. All included images are from the original paper.

² Department of Bioengineering, Imperial College London, UK

Chapter Abstract

Automated source separation algorithms have become a central tool in neuroengineering and neuroscience, where they are used to decompose neurophysiological signal into its constituent spiking sources. However, in noisy or highly multivariate recordings these decomposition techniques often make a large number of errors. Such mistakes degrade online human-machine interfacing methods and require costly post-hoc manual cleaning in the offline setting. In this paper we propose an automated error correction methodology using a deep metric learning framework, generating embedding spaces in which spiking events can be both identified and assigned to their respective sources. Furthermore, we investigate the relative ability of different DML techniques to preserve the intra-class semantic structure needed to identify incorrect class labels in neurophysiological time series. Motivated by this analysis, we propose locality sensitive mining, an easily implemented sampling-based augmentation to typical DML losses which substantially improves the local semantic structure of the embedding space. We demonstrate the utility of this method to generate embedding spaces which can be used to automatically identify incorrectly-labelled spiking events with high accuracy.

4.1 Introduction

Neurophysiological signals can generally be characterised as an additive mixture of repeating events from different sources, such as the motor unit activation potential in electromyography or the spike potential in microelectrode cortical recordings[24][108]. The ensemble of these activation events constitute neural codes that can provide direct insights into a neurological system of interest[109][16][46]. Consequently, the extraction of individual sources from noisy signal by blind source separation algorithms has long been a major focus in computational neuroscience[16][110][111][112]. Modern BSS algorithms leverage highly multivariate data, such as the output from high-density electrode arrays[113][114], to identify the contributions of many individual spiking sources[16][115][48]. These sources also provide an extremely clean control signal for human-machine interfacing applications when compared to the original bulk signal[46][116][117].

Despite these successes, BSS algorithms frequently make mistakes when labelling source activations, which can be difficult to identify and correct automatically[118][14]. As a result, applications that use neurophysiological data still broadly rely on bulk signal[119][120][121]. In offline decomposition, a degree of manual or semi-automated post-hoc cleaning is commonly employed, often using additional knowledge about the system of interest, for example the temporal statistics of the sources[122]. The nature of this manual error correction generally relates to the mixing system of interest, for example, intracortical and intramuscular EMG decompositions generally require post-hoc examination of source classes[112][5], whilst surface EMG decompositions also require further inspection of individual neural activations[123]. However, whilst accurate, manual cleaning is constrained to offline methods and is an extremely time-consuming process[124]. In response, modern source separation pipelines are

increasingly using additional automated post-processing steps[14][122][125]. A contemporary direction is to use the noisy BSS-derived labels to train a neural network classifier, which can then outperform the noisy label it was trained on[43][78][126]. However, such supervised deep learning approaches generally suffer from a critical threshold where the initial label noise overwhelms the ability of the model to self-correct[127].

A different approach to leveraging deep learning for neurophysiological source separation is through contrastive methods such as deep metric learning, in which a model is trained to learn the relative similarity or dissimilarity between data samples rather than to directly classify them[128][129]. In this formulation, the neural network acts as an efficient featurizer, taking high-dimensional neurophysiological input and extracting a low-dimensional embedding in which individual spiking events are easily assigned to their respective sources. In recent years DML has seen particular success in person re-identification tasks[130][131], medical image classification[132][133] and digital pathology[134][135]. In the theoretical domain, problems with optimisation stability and computational inefficiency have been largely solved through a combination of sampling strategies[136][137][138] and better distance metrics[128][139][140][141]. DML is attractive in the context of spiking events, which are stable over time given fixed conditions. This means that for a well trained model, each spiking event from the same source should cluster in the same location in the embedding space, with the DML model theoretically only needing to remove the distortions caused by noise and superposition events. DML also has lower data demands and a better tolerance for class-imbalance when compared to typical neural network-based classifiers[142], both of which are common issues in neurophysiological time series processing.

In this paper we have two main objectives for a DML-trained featurizer. Firstly, we

want neural events from different sources to map to different regions of the embedding space so they can be discriminated, i.e. we want good inter-class variance. Secondly, we want neural events from the same source to preserve some intra-class structure for the purposes of identifying label noise. This second objective is complicated by the fact that maintaining a degree of intra-class variance is not usually the focus of DML methods, which are generally more interested in inter-class separation[143]. Current losses tend to incentivise the model to ignore intra-class semantic differences and collapse the embedding down to a tight cluster [138]. To mitigate this, we propose a simple method of augmenting DML training to prevent this happening, allowing the embedding to be used for more than just class separation.

In summary, the main contributions of this paper are as follows:

1. We robustly demonstrate that DML is an effective method of building a neural network featurizer for source separation of neurophysiological signals. Using sEMG signal, we show that the activations from different sources clearly cluster, making class discrimination by simple clustering methods trivial.
2. We implement a number of popular DML methods such as N-pair loss and angular loss [139][141], demonstrating that they generate embedding spaces with excellent class discriminability, but poor local semantic structure. We show that more contemporary methods which preserve such structure can be used to identify outliers within an artificially corrupted neurophysiological signal.
3. We propose locality sensitive mining (LSM), an easily-implemented sampling-based method of maintaining intra-class structure that can be implemented in a variety of different DML paradigms. We go on to show that for neurophysiological signal LSM

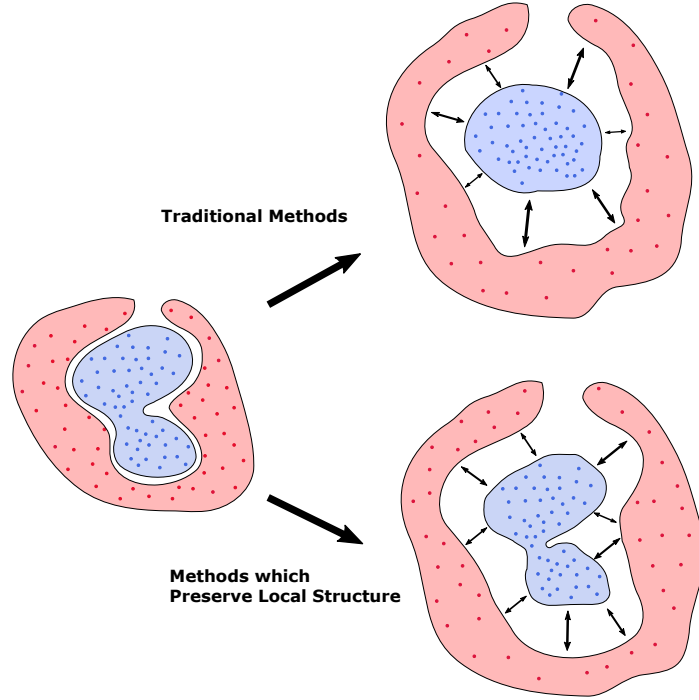


Figure 4.1.1: The effect of traditional deep metric learning losses versus those designed to preserve local semantic structure. Traditional methods tend to obliterate intra-class variance in the embedding of samples from the same class (in blue), whilst recent directions instead preserve this, while maintaining good separation from samples belonging to different classes (in red).

outperforms other contemporary methods of preserving local semantic structure such as Multi-level Distance Regularisation (MDR)[143], whilst having a very simple implementation that can be added to many different DML losses.

4. For the first time we are aware of in the literature, we show that DML methods which maintain intra-class semantic structure have clear practical utility for identifying incorrectly labelled outliers in the class clusters. As an example, we leverage this to build a self-correcting source separation pipeline for neural spiking signals, which we call the DeepDecomp pipeline. This encompasses the process of identifying and discarding false positives, followed by identifying and including false negatives.

4.2 Related Work

DML has classically been employed in discrete high-dimensional data types such as images, rather than in the time series domain. This has not been due to a lack of theoretical grounding[144][145][146], but relates more to the difficulty of building discrete class pairings for signals when components have variable temporal dynamics, which may explain the lack of DML methods for neurophysiological signal processing in the literature. However, when the independent factors in the generative process have relatively short and stereotyped responses, such as spiking neurons, it becomes straightforward to break the signal into windows centred on each activation. In this case, the problem becomes similar to face re-identification, where the neural network needs to learn an invariance to confounding factors when bringing images of the same face closer together in the embedding space[130][131][147]. There has also been some work in face re-identification to make training robust to label noise, using external models to build metrics of label quality[148][149][150][151]. Unfortunately, these methods are generally reliant on an extremely large amount of data to be effective, which precludes their use in most neurophysiological datasets.

An alternative solution to the problem of label noise is to preserve a richer intra-class embedding, such that semantic similarity or dissimilarity between samples from the same class is better preserved, allowing identification of outliers. Early work on DML losses generally either ignored intra-class variance or actively sought to reduce it[140][152]. Some proxy-based losses, which pull samples towards shared class proxies rather than other in-class samples[153][154], have been designed in part with the intention of reducing overfitting by preventing intra-class collapse[155][156]. There has also been some work in preserving intra-class characteristics using generative models such as variational autoencoders [157][158], with the main objective

of improving generalisation performance by excluding features that are not shared across a class when making embedding decisions.

More recently there have been some explicit attempts to preserve local semantic structure in the embedding space, as illustrated in figure 4.1.1. DML with self-supervised ranking uses an auxiliary term based on self-supervised learning[143], using a set of transform functions to augment samples to different degrees, such that more perturbed samples are embedded further away from the original image. However, this requires domain knowledge when sculpting the transform functions, for example the authors perturb image scale, viewpoint and colour in their task of bird identification. Selecting similar transformations for neural signals would not be a simple task. An alternative approach is multi-level distance regularisation, which also adds an auxiliary loss term, albeit one which aims to prevent class collapse by using proxies set at multiple distances from each sample[159].

4.3 Preliminaries

4.3.1 Deep Metric Learning

The basic aim of DML is to train a neural network to map a sample taken from one of C classes to an embedding vector $\mathbf{x}$, such that for an arbitrarily selected anchor embedding $\mathbf{x}_a$, samples from the same class $\mathbf{x}_p$ are embedded closer than samples from a different class $\mathbf{x}_n$, as measured by some distance metric $\mathbf{D}$. $\mathbf{D}$ can be a number of different metrics, such as the Euclidean distance, cosine similarity or symmetrized Kullback–Leibler divergence[160]. Commonly the loss function is formulated in terms of a relative distance between positive pairs (samples from the same class) and negative pairs (samples from different classes), such

that:

$$\mathbf{D}(\mathbf{x}_a, \mathbf{x}_p) + m \leq \mathbf{D}(\mathbf{x}_a, \mathbf{x}_n) \quad (4.3.1)$$

where m is a margin term that specifies the objective inter-class separation.

After a small amount of optimisation, the bulk of negative pairs will be further away than positive pairs, making most training examples in subsequent batches uninformative[136]. This can be partially mitigated by mining strategies; calculating the DML loss using only pairings from each batch selected using some heuristic based on the embedding space[161], for example selecting only pairings where $\mathbf{x}_a$ is further away from $\mathbf{x}_p$ than $\mathbf{x}_n$. The metric used for mining is usually the same as used for calculating the metric loss. A related approach is to use multiple negative pairings N for each anchor term, as proposed originally in the N-pair loss[139]:

$$L_{NPAIR} = \frac{1}{M} \sum_{\mathbf{x}_a \in B} \log \left[1 + \sum_{\mathbf{x}_n \in B \setminus (\mathbf{x}_a, \mathbf{x}_p)}^N \exp(f_{a,p,n}) \right] \quad (4.3.2)$$

where $f_{a,p,n}$ is the DML loss and B is a batch of size M .

4.3.2 Locality Sensitive Mining

Whilst current methods of selecting pairings from B are suitable if the objective is to maximise inter-class distance within the embedding space, they have the side effect of penalising local semantic structure within a class[153]. When positive pairings are selected randomly, the network will generally learn to embed all samples from a particular class into a dense point[156].

To preserve local semantic structure, we instead propose a new mining procedure, locality sensitive mining (LSM), using a top-k algorithm to select the k closest (based on the same distance metric) $\mathbf{x}_n$ to $\mathbf{x}_a$ that belong to a different class and using only the closest $\mathbf{x}_p$. GPU implementations of top-k algorithms have become extremely efficient in recent years, due to their increasing use within machine learning applications[162]. In the N-pair formulation this mining approach can be written as:

$$l_{LSM} = \frac{1}{M} \sum_{\mathbf{x}_a \in B} \sum_{\mathbf{x}_p \in B \setminus (\mathbf{x}_a, \mathbf{x}_n)} \mathbf{1}_{\mathbf{x}_p \in \Upsilon} \log [1 + g] \quad (4.3.3)$$

$$g = \sum_{\mathbf{x}_n \in B \setminus (\mathbf{x}_a, \mathbf{x}_p)} \mathbf{1}_{\mathbf{x}_p \in \Theta} \exp(f_{a,p,n})$$

where $\Upsilon = \operatorname{argmax}(\angle(\mathbf{x}_a, \mathbf{x}_p))$ is the argmax set of the pairwise cosine similarity $\angle$ between $\mathbf{x}_a$ and its associated set $\mathbf{x}_p$ in B and $\Theta = \operatorname{topk}(\angle(\mathbf{x}_a, \mathbf{x}_n))$ is the top-k values of the ordered pairwise cosine similarity $\angle$ between $\mathbf{x}_a$ and its associated set $\mathbf{x}_n$ in B . $\mathbf{1}$ is the indicator function.

Like other mining procedures, LSM can be combined with many DML losses. As examples, we selected N-pair loss combined with the popular angular loss as a distance metric[141].

4.3.3 Angular Loss

Angular loss is a stable geometric reformulation of the angular distance metric, which constructs a right-angled triangle with $\mathbf{x}_n$ and the midpoint between $\mathbf{x}_a$ and $\mathbf{x}_p$, with the final vertex being the point on the semicircle joining $\mathbf{x}_a$ and $\mathbf{x}_p$ [141]. By dropping constant terms, this geometric relationship can be used as the DML loss $f_{a,p,n}$ in equation 2, expressed as:

$$f_{a,p,n} = 4\tan^2\alpha(\mathbf{x}_a + \mathbf{x}_p)^T \mathbf{x}_n - 2(1 + \tan^2\alpha)\mathbf{x}_a^T \mathbf{x}_p \quad (4.3.4)$$

where α is an angle in radians which sets the upper accepted bounds of the loss, analogous to m in equation 1.

4.3.4 Cross Entropy Loss

A major difficulty in optimising a model to detect spiking events is that spikes are relatively rare, meaning the data has a strong class imbalance towards windows with no activity[43]. This is particularly problematic in DML as the losses generally have local optima of mapping all samples to a single location in the embedding space. We found that the addition of a small auxiliary cross-entropy term with temperature was useful in avoiding this trivial solution:

$$l_C = -\frac{1}{M} \sum_i^M \sum_c^C \mathbf{1}_{y_i \in C_c} \log \frac{\exp\left[\frac{1}{\gamma} \mathbf{z}_i\right]}{\sum_j^C \exp\left[\frac{1}{\gamma} \mathbf{z}_j\right]} \quad (4.3.5)$$

where $\mathbf{z} = \frac{\mathbf{W}}{\|\mathbf{W}\|_2} \mathbf{x}$ and $\mathbf{W}$ is a trainable matrix that compresses the embedding vector down to a dimension C vector for comparison with the one-hot encoded class labels. The impact of the cross-entropy term was weighted by a coefficient such that:

$$l = l_{DML} + \tau l_C \quad (4.3.6)$$

4.3.5 Neurophysiological Signal Decomposition

The timestamps used by DeepDecomp can be generated using a wide variety of manual and automated processes, however, for this study the gradient convolution kernel compensation

algorithm was selected due to its strong performance in sEMG signal decomposition[9][13]. Optimised sources were converted to timestamps using a linear threshold or a two-class k means clustering algorithm.

4.4 Validation Methodology

4.4.1 Evaluated Methods

We compared a number of different DML methods, examining the utility of their respective embedding spaces for identifying two types of label noise and classifying unseen neural activations. The methods studied were as follows:

N-pair Loss (NL)[139]: The original N-pair loss using Euclidean distance, where each sample in a batch is paired with one randomly selected in-batch positive and all in-batch negatives. A small auxilliary cross-entropy term, l_C in equation 5, was added to stabilise training, with both γ and τ set to 0.1.

Angular Loss with NL (AL)[141]: The N-pair loss (with the same auxiliary cross-entropy term), but using angular loss instead of euclidean distance as the distance metric. α to 0.25 radians in both the cleaning and refitting stages. It should be noted that this method also acts as an ablation of the LSM component of training.

AL with Locality Sensitive Mining (LSM): The same loss as AL (with the same hyperparameters), but using the proposed mining approach instead. k was set to 5 throughout, which was selected after a hyperparameter study also detailed in this paper.

Multi-level Distance Regularisation (MDR)[143]: We also compare our approach with

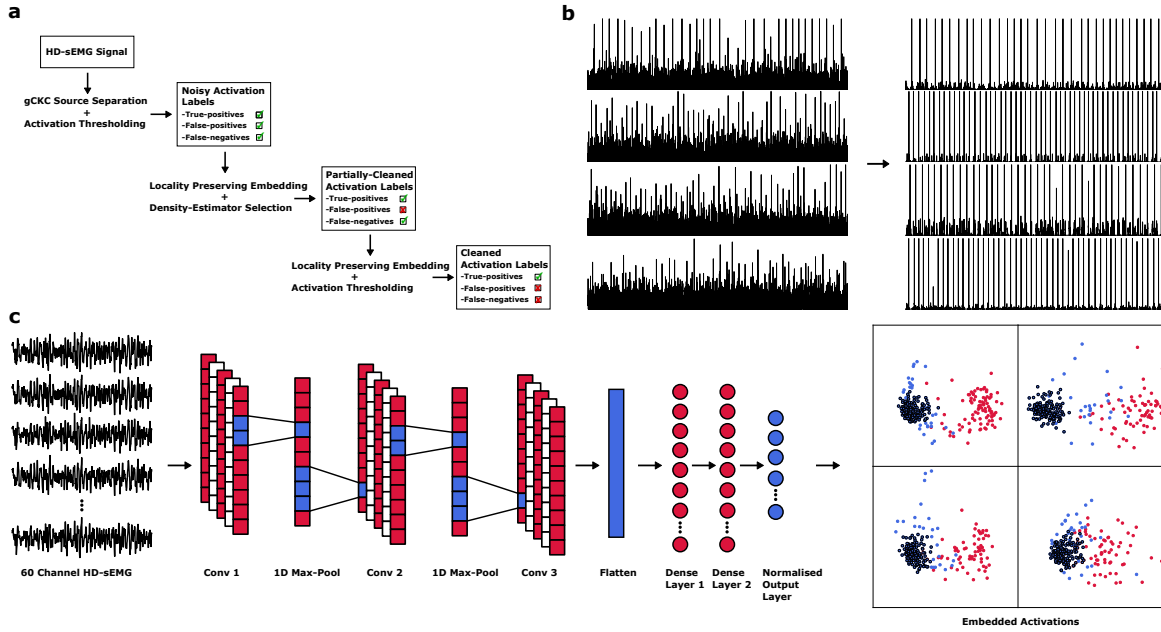


Figure 4.4.1: **a** shows DeepDecomp, an example pipeline by which the noisy activation labels found from source-separating the high-density surface electromyography signal are cleaned. The model is trained twice - a cleaning phase to find the false positive labels and a refitting phase to find the false negatives. After the refitting phase the predicted class activity is much cleaner than that of the original source separation algorithm, as seen in **b**. **c** shows the convolutional neural network trained by a deep metric learning loss to find neural activations. Windows of neurophysiological time series are embedded into a low-dimensional space which can be used to source separate and, if a method that preserves local semantic structure is used, to clean a noisy label set.

the recent MDR, which also aims to preserve intra-class variance, albeit mainly from the perspective of improving generalisation performance.

Cross-entropy (CE): As a baseline we also included a classifier trained using cross-entropy, i.e. purely l_C in equation 5 with both γ and τ set to 1. The embedding dimension is simply the output of the layer before $\mathbf{W}$.

4.4.2 Experiments

In experiment 1 we evaluated the ability of the models to clean a label set corrupted by feature-dependent noise, where label flipping probability is related to its associated features[163].

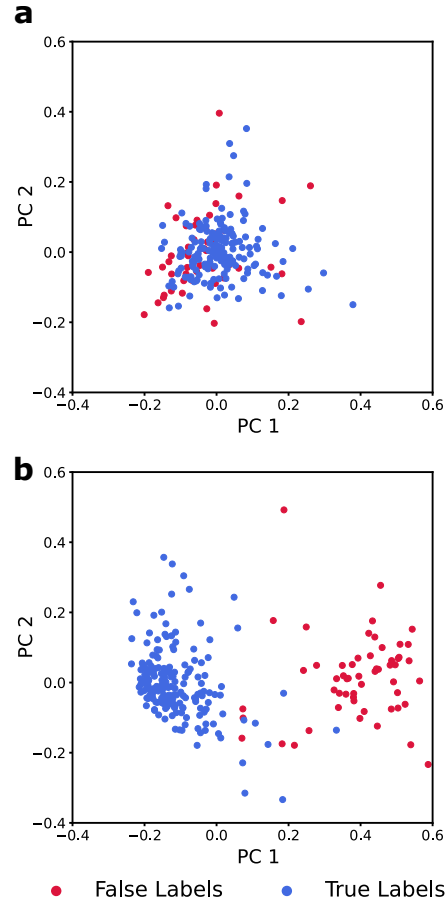


Figure 4.4.2: The effect of two different deep metric learning losses on the embedding space for two units as shown by the first and second principle components for 40% corruption with feature-dependent label noise. **a** shows the effect of using using the original random sampling method of angular loss, leading to all intra-class embeddings contracting down to a point. In **b** the same optimisation was run again using a DML loss that preserves local structure, creating an embedding space in which the true labels cluster away from the false labels.

In the context of source-separated sEMG, this most commonly occurs as a false-positive, where a separation vector incorrectly assigns a high probability of an in-class MUAP being present when it is not, i.e. a noise class or other MU class label is flipped to the MU class of interest. We define a label as “true” when it is a timestamp that contains a MU activation that has been labelled as the member of the correct MU class, and “false” otherwise. To simulate this effect, we corrupted the label set by generating an artificially-noisy separation vector for each MU class by randomly selecting 15 MUAP labels from that class and using the average of the associated extended sEMG vectors to generate a linear minimum mean square error prediction on the extended sEMG matrix. A two-class k-means clustering algorithm was then used to parameterise a linear threshold to find activations, creating a label set with a high degree of feature-dependent noise. Five levels of increasing difficulty were generated by taking an amount of false positives corresponding to 10/20/30/40/50% of the number of true labels, selected at random from the set of false positives.

In experiment 2 the models were instead evaluated on class-dependent label noise, when the probability of a label flipping to another class is stable across all labels in the class[163]. As for most neural spiking signals, in sEMG source separation this error generally occurs when the separation vectors are very similar, usually due to similar MUAP waveform shapes between two MU classes. This can be simulated by transferring a percentage of labels to a similar MU class. This was done by first averaging the MUAPs of each MU class and then cross-correlating these averages with the average MUAP of every other class in the recording, with 10/20/30/40% of the class labels transferred to the class with the highest value. If labels had already been transferred to the closest class then the next closest class was selected until all classes had had label transfers. A maximum label corruption of 40% was used to preserve

the concept of a majority true and minority false class.

After training with the artificially corrupted label sets, methods which successfully preserved intra-class variance should have a dense central cluster due to the methods of label corruption. To avoid bias from manual selection and to demonstrate a fully-automated pipeline, we elected to automatically assign samples to this cluster using a simple density-estimator. As the cleaning process could potentially bias the label set by incorrectly removing in-class outliers, we also retrained the models with the cleaned sets and used the models to find unlabelled spiking events. This also allowed demonstration of the generalisation performance of the models trained with the different loss functions. Finally we explored the impact of the k hyperparameter within LSM by rerunning experiment 1 with multiple values of k .

As a demonstration of the practical utility of the proposed method for building a self-correcting source separation pipeline for neural spiking signals, we give an example pipeline, DeepDecomp, which automate the cleaning of the output of a noisy gCKC decomposition of sEMG signal (fig. 4.4.1a). By preserving local semantic structure, noisy activation label sets can be accurately cleaned (fig. 4.4.1b).

4.4.3 Model and Training

To convert the source-separated sEMG signal into labelled windows, first each channel of the sEMG signal was standardised by z-scoring and then cut into overlapping 80-sample wide windows at a stride of 1. Each window was then labelled by reference to the predicted source activity at the final sample of the window. This meant the bulk of windows were labelled as part of the inactive class due to the sparse nature of motor neuron spiking. Due to this serious class imbalance, each minibatch was created from the entirety of the windows labelled

as containing a motor neuron spike, with an additional 256 samples randomly selected from the inactive class. Each class assignment was then converted to a one-hot representation, the bulk of which had only one class active at any one time, although rarely two activations would occur simultaneously on the same time-point. As the richness of the intra-class embedding of the inactive class windows was not of any great concern, the embeddings of these windows were not used as anchor samples when calculating the DML component of the tested methods, although they were used as negative samples and in the calculation of l_C .

A convolutional neural network implemented using the Tensorflow machine learning library in Python was used as the embedding model (fig. 4.4.1c). Grid search optimisation was used to select specific model architecture hyperparameters. Convolution steps used a 1D 3-sample wide kernel, with 32 filters and a drop-out of 0.2. 1D max-pooling was completed with 2-sample wide kernels. Each densely-connected layers had 64 neurons and a drop-out percentage of 0.5 during training. Both the convolution and densely-connected layers used ReLU activation functions. Finally the output of the last densely-connected layer was densely-connected to a bias and activation-free embedding layer of 8 neurons-wide, which was then divided by its L2 norm. This was an intentionally low-dimension embedding compared to standard DML due to the desire to avoid dimensionality issues during the clustering steps in the refitting phase. The additional matrix $\mathbf{W}$ used in the categorical cross-entropy was initialised with truncated normal noise, whilst the weights of the neural network layers was initialised by glorot uniform. The Adam optimisation algorithm at a learning rate of 0.001 was then used to train the model over 500 epochs for both cleaning and refitting stages.

When the model was used to find neural activations, the cosine similarity of each sample to the average embedding vectors of classes was calculated, with the spiking activations assigned

Table 4.1: Cleaning Results for Different Levels of Feature-dependent Label Noise
Level of Added Label Noise

	10%		20%		30%		40%		50%	
Method	True and False Labels after Cleaning (% of Original Set)									
	True	False	True	False	True	False	True	False	True	False
CE	90.2	7.9	88.8	16.2	89.3	25.1	90.9	32.6	79.7	35.6
NL[139]	92.8	7.2	90.3	14.8	87.8	20.0	86.9	24.4	77.6	25.8
AL[141]	66.7	2.2	58.4	5.7	54.6	8.0	45.1	8.0	42.0	10.8
MDR[143]	72.1	1.4	68.6	1.6	65.7	2.3	62.0	1.9	59.9	3.2
LSM	84.1	1.0	78.3	1.0	80.3	1.5	76.6	1.4	74.1	2.3

to MUs by way of a two-class k-means clustering algorithm. These activation labels could then be compared to the pre-corrupted data using the rate of agreement (RoA) metric[24], a percentage defined as the number of true positive matches divided by the total number of true positives, false positives and false negatives. When the model was instead used to clean the corrupted label set, a simple density estimator was used. First a local scale value v was estimated by finding the mean cosine similarity of the each embedding vector to its 20 nearest neighbours and taking a median of this value across all vectors. For each label the number of other labels with a cosine similarity more than v was found and the label with the highest number of neighbours was selected as the centre of the cluster. All labels within a cosine similarity higher than v were then added to the refitting training set. This simple approach was generally adequate for quickly finding the densest region of the embedding space, which was usually the cluster of true labels.

4.4.4 Experimental Dataset

The sEMG data set consisted of 20-second recordings taken from the dominant tibialis anterior muscle of 10 men performing an isometric contraction at 15% of maximal force, previously used to validate source separation techniques[62]. Maximal contraction was defined as the

Table 4.2: Cleaning Results for Different Levels of Class-dependent Label Noise

Method	Level of Added Label Noise							
	10%		20%		30%		40%	
	True and False Labels after Cleaning (% of Original Set)							
	True	False	True	False	True	False	True	False
CE	79.8	7.9	64.5	17.7	58.1	25.4	54.5	33.5
NL[139]	82.5	7.1	64.0	13.5	57.9	20.9	49.7	30.1
AL[141]	61.7	3.0	45.5	9.3	41.8	13.0	33.5	20.0
MDR[143]	63.1	0.8	48.1	1.6	43.9	1.8	36.4	2.2
LSM	86.4	0.8	75.4	0.9	60.9	0.9	53.1	1.0

mean force of three 5-s maximal contractions separated by 3 min of rest, with force sampled at 2048Hz by load cells mounted on an isometric brace. Force feedback was provided to the participants by an oscilloscope. The signal from a monopolar 12×5 electrode array placed over the main muscle innervation zone was sampled at 2048Hz having been band-pass filtered at 10-500 Hz. Gradient convolution kernel compensation with an additional k-means source refinement step was implemented using the tensorflow machine learning package[14][13]. As the label set was to be artificially corrupted it was important that the original be as noise-free as possible, so additional post hoc steps were taken to maximise the likelihood that the timestamps were correct. Sources were manually cleaned by examining interspike intervals and the source-to-noise ratio of each activation. An additional step of validating decomposition accuracy was implemented by comparing the sources to those found using the DEMUSE source-separation software package[9][13], with source cleaning completed by a different trained operator.

4.5 Results

4.5.1 Feature-dependent Label Noise

In experiment 1, which tested the effect of feature-dependent label noise by simulating noisy separation vectors, the LSM-trained network generated an embedding space with dense clusters for each class corresponding to the true labels. Surrounding each cluster is a large sparse periphery of false labels with no apparent structure, the expected result as these false samples shared fewer features. In contrast, the embedding spaces generated by training with CE, NL and AL gave a more uniform single cluster for each class (with some distant outliers for CE). The utility of these different embedding spaces for identifying false labels is particularly clear when LSM is compared to AL using a two-dimensional principal component space (fig. 4.4.2).

Table 1 shows the cleaning results when selecting only samples from the highest density region of a class embedding. The LSM-trained network generated an embedding space with utility for removing false labels even at the maximum tested value of 50% of total correct values, with a median post-cleaning false label retention of 2.3% of the total correct labels in the class. The number of true labels lost during the cleaning process fell as the pre-cleaning percentage of false labels increased, but even at the highest false label percentage tested, a median of 74.1% of the true values were still retained.

MDR also performed well at preserving some local semantic structure and so the embedding space had some utility for identifying noisy labels, although it was slightly outperformed by LSM. In contrast, the networks trained with CE, AL and NL did not generate embedding spaces suitable for label cleaning. AL and MDR generally produced much more distributed

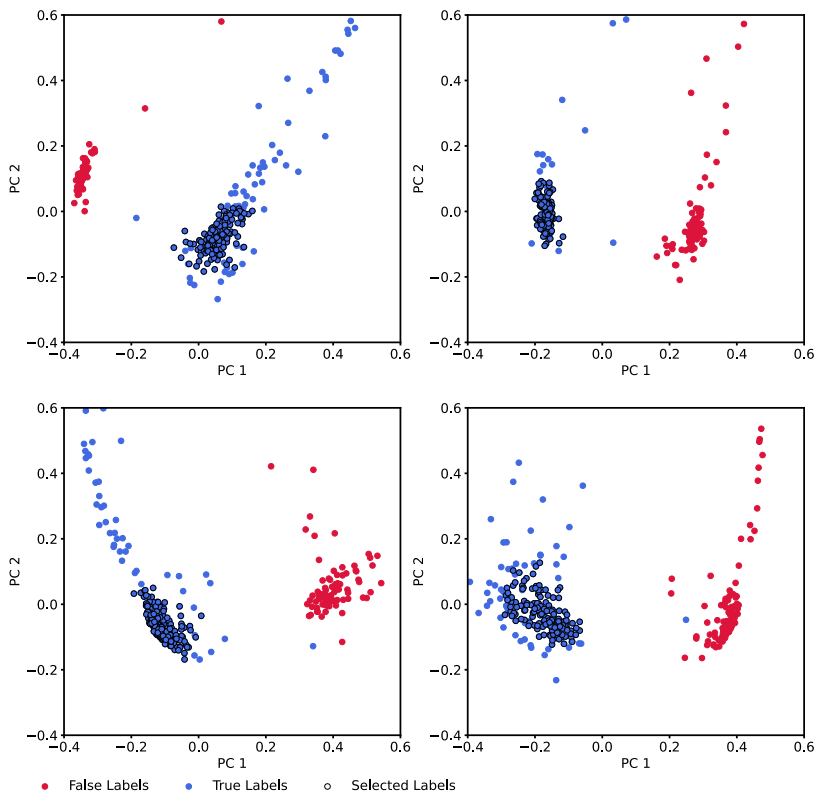


Figure 4.5.1: Principal component plot of the embedded samples from four classes with added class-dependent label noise. The samples selected automatically for the refitting phase have been circled. The false labels have shared features as they come from the same class. This results in two tight clusters for both true and false labels, however, they are still clearly separable.

clusters, meaning less samples in total, true or false, were selected by the density estimator.

4.5.2 Class-dependent Label Noise

In experiment 2, when labels were randomly flipped to the MU class with the closes average MUAP shape, LSM again generated embedding spaces with clear separation between true and false separation. However, unlike in the first experiment, the false labels formed a second distinct cluster within the embedding space (fig. 4.5.1). As the true label cluster always had more values, it was still clearly identified by the density estimator.

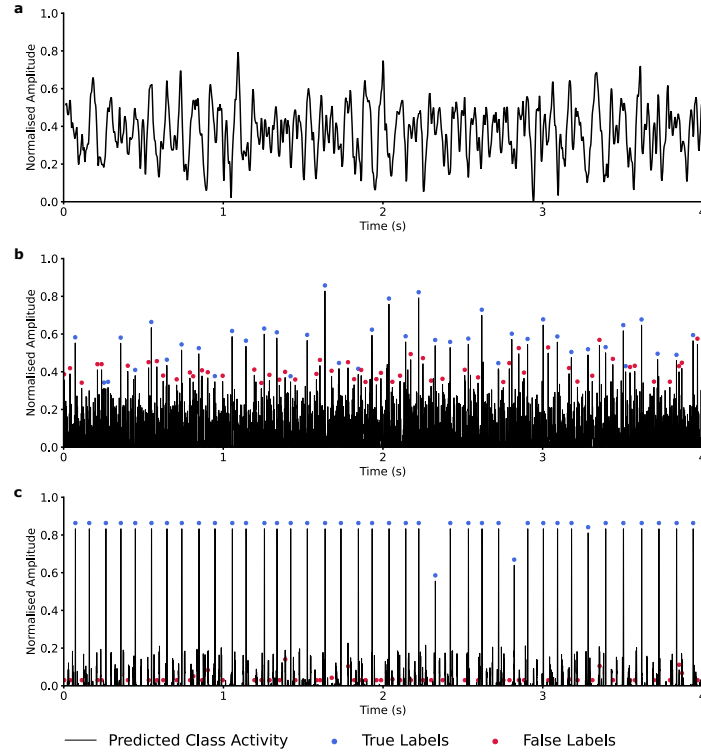


Figure 4.5.2: A single channel of unprocessed sEMG and the post-decomposition predicted activity of a single class before and after cleaning and refitting, with true and false labels. **a** demonstrates the degree of complex superposition inherent to sEMG signal as opposed to cleaner recordings such as those from intracortical sources. A linear separation filter based on an average of only 15 labels is applied to the signal to generate **b**, which is consequently extremely noisy, simulating a poorly optimised filter. A number of false positives corresponding to 50% of the number of true class labels has been selected. After the cleaning and refitting phases of DeepDecomp the spiking motor neuron activity in **c** is clearly identifiable, whilst incorrect labels have been suppressed.

Table 2 gives the cleaning results for class-dependent label noise across the different tested methods. As in experiment 1, the LSM-trained model generated an embedding space that allowed the density estimator to identify almost all false labels. Even at a 40% transfer the median post-cleaning false label retention was only 1.0% of the total correct labels in the class. MDR also performed quite well, albeit with a slightly higher false label burden than LSM.

As true labels were lost both to the initial transfer to other classes and to the cleaning phase, far fewer were retained in the post-cleaning dataset than in experiment 1 and would need to be recovered in the refitting stage. Once again, the CE, NL and AL trained models generated embedding spaces that were not useful for cleaning, and AL and MDR again generating a looser embedding with the consequence of less samples selected.

4.5.3 Rediscovering Unlabelled Activations

An important requirement if the cleaned label set is to be useful is that the cleaning process does not overly bias against true labels that are lost at this stage, making them difficult to recover. Lost labels tend to be more peripheral in the cluster, meaning their MUAP shapes are likely to be less similar to the MU class average, potentially due to superposition with a MUAP from a different class or due to a noise artefact. False negatives are also still used for training, but are labelled inappropriately, with a possibly detrimental effect of the model to generalise. To demonstrate that neither of these potential problems actually impacted training, after the cleaning stage of both experiment 1 and 2, the model was refitted with the LSM-cleaned label set. As an additional comparison, we also refitted using the same label set, but with the other methods compared in this study.

For both experiments the predicted activity after running the DeepDecomp pipeline was generally both sparse and clean, with MUAPs easily identifiable using all methods tested. These labels were compared to the original data using the rate of agreement metric, with results given in table 3. LSM and MDR tended to outperform the other methods tested, most likely due to a better ability to generalise that has been noted in DML losses that preserve semantic information[143]. Most striking was the ability of the DeepDecomp pipeline to recover the original label sets in extreme degrees of label noise, as demonstrated visually in figure 4.5.2.

Table 4.3: Refitting Results for Different Levels of Label Noise

Label Noise Type	Method	Level of Added Label Noise				
		10%	20%	30%	40%	50%
		Rate of Agreement (%)				
Feature Dependent	CE	92.1	92.3	90.8	90.6	89.1
	NL[139]	94.1	94.3	91.0	92.3	92.3
	AL[141]	93.7	93.7	89.4	89.2	91.5
	MDR[143]	94.6	94.0	93.4	93.6	91.7
	LSM	94.5	94.5	93.5	93.8	92.0
Class Dependent	CE	95.3	92.8	91.0	89.3	
	NL[139]	97.7	95.4	91.5	89.7	
	AL[141]	95.3	93.4	91.2	90.2	
	MDR[143]	98.0	96.8	94.5	93.8	
	LSM	98.2	96.8	95.7	94.0	

4.5.4 Impact of the k Hyperparameter

Two main trends emerged when the cleaning component of experiment 1 was repeated using different values of the k hyperparameter in the LSM method, which controls the number of top- k negatives which are used to build the loss (fig. 4.5.3). As k increases the intra-class embedding becomes increasingly dense, resulting in the density estimator selecting a greater number of samples. At the same time, the embedding begins to lose its intra-class variance,

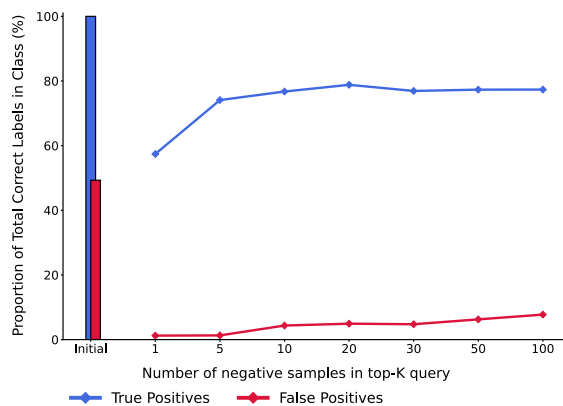


Figure 4.5.3: Impact of the number of top-k negative samples used in locality sensitive mining during the cleaning phase, after the data labels were corrupted with feature-dependent label noise. As k rises, more true positives are densely clustered in the embedding space, but the network begins to lose its intra-class variance, causing the density estimator to fail. A k of 5 was found to be optimal for maximising the number of true positives whilst preventing class collapse.

resulting in increasing numbers of false negatives being incorrectly assigned as true. This leads to an optimal value of k being 5 if the main objective is to maximise the likelihood of identifying false negatives. Its worth noting that a k of 1 is functionally similar to an easy-negative loss[138], although this is quite inefficient with respect to losing true positives to the cleaning process.

4.6 Conclusion

In this study we demonstrate that a DML pipeline which preserves local semantic structure can embed high-dimensional neurophysiological signal into a low-dimensional space that allows accurate identification of incorrectly labelled activation events. Furthermore, we present locality sensitive mining, a sampling-based augmentation to DML losses which preserves such semantic structure. By using artificially corrupted sEMG data, we show that this simple change can outperform other contemporary methods of preserving intra-class variance, mak-

ing possible a practical pipeline for cleaning noisy source separated time series data. As an example, we created DeepDecomp, a pipeline which utilises LSM-augmented DML to clean heavily-corrupted sEMG decomposition data over two passes. Importantly the model was still able to perform even when the source of label corruption is class or even feature dependent, an important consideration in neurophysiological signal where mistakes often occur due correlated effects such as source superposition.

Although this study focused on source-separated sEMG signal, it is important to emphasise the broader applicability of this approach to any imperfectly-labelled neurophysiological time series data characterised by repeating events. Whilst the study focused specifically on action potentials, the proposed methodology could also be used for pattern recognition in bulk neurophysiological signal, for example by supplementing contemporary prosthetics and exoskeletons[164][165]. Additionally, the labelling process need not be by a BSS algorithm. For example, a DeepDecomp-like pipeline could be applied to a dataset for which only a small component of the data has been manually labelled by an expert-operator, recovering the rest of the labels accurately. In this way, the proposed approach can be viewed as a minimally-supervised method for decomposing neurophysiological time series into individual cell activities.

Chapter 5

Compensating for Non-Stationarities in EMG with Deep Learning¹

¹This chapter is based on *HarmonICA: Neural Nonstationarity Correction and Source Separation for Motor Neuron Interfaces*, which is currently under consideration as a paper in the Thirty-Eighth Annual Conference on Neural Information Processing Systems with co-authors Agnese Grison², Irene Mendez Guerra², Pranav Mamidanna², Shihan Ma², Silvia Muceli³ and Dario Farina². Some minor changes have been made to better suit the thesis structure. As the paper presents a solution to the non-stationary version of the EMG forward model, the theory section has been largely unchanged, which means there is a small amount of duplication of previous theory. The included image is from the original paper.

² Department of Bioengineering, Imperial College London, UK

³ Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden

Chapter Abstract

A major outstanding problem when interfacing with spinal motor neurons is how to accurately compensate for non-stationary effects in the signal during source separation routines, particularly when they cannot be estimated in advance. This forces current systems to instead use undifferentiated bulk signal, which limits the potential degrees of freedom for control. In this study we propose a potential solution, using an unsupervised learning algorithm to blindly correct for the effects of latent processes which drive the signal non-stationarities. We implement this methodology within the theoretical framework of a quasilinear version of independent component analysis. The proposed design, HarmonICA, sidesteps the identifiability problems of nonlinear ICA, allowing for equivalent predictability to linear ICA whilst retaining the ability to learn complex nonlinear relationships between non-stationary latents and their effects on the signal. We test HarmonICA on both invasive and non-invasive recordings both simulated and real, demonstrating an ability to blindly compensate for the non-stationary effects specific to each, and thus to significantly enhance the quality of a source separation routine.

5.1 Introduction

Great strides have been made in the miniaturisation and reliability of high density electrode arrays for capturing surface and intramuscular electromyography (EMG) signals[166, 167]. With source separation algorithms, these recordings give direct access to the spinal motor neuron activity and so provide an excellent target for human computer interfacing applications. Reliable access to individual motor neurons offers the potential for a new generation of intuitive prosthetics and may even allow the extra degrees-of-freedom needed to control additional robotic limbs[168, 169, 170].

Despite these opportunities, a significant remaining obstacle to the development of interfaces based on individual neural activity is the brittleness of current decomposition methods to non-stationarities in the signal, which limits the majority of spinal interfaces to inference at the population level. These are the result of changes in the spatial position of the electrodes relative to the neuron and fluctuations in the volume conductor through which the biopotential is propagating. The former, called drift, can be estimated and corrected by interpolation methods for small movements in the plane of the electrode array, which is the basis of modern intracortical spike sorting algorithms[171, 172].

Outside of this specific scenario, when there is drift in multiple non-planar directions or volume conductor change, it remains extremely difficult to compensate for the complex resultant variation in the action potential waveforms over time during the source separation routine. There have been some recent successes using sliding windows and *post hoc* correction to handle subtle variation over time[173, 174], but these approaches often fail when neurons are not consistently active in all windows or the non-stationary changes occur too rapidly. This has a strong limiting effect on experimental design, for example almost all spinal neuron

recordings made using surface EMG restrict the subject to a fixed isometric position to prevent volume conductor variation.

In this work we propose HarmonICA, a deep learning model for source separation of non-stationary neurophysiological time series using quasi-linear independent component analysis (ICA). By using an alternating backpropagation strategy to revert to a linear model during source estimation, we sidestep the identifiability problems of true non-linear blind source separation. This gives the model the predictability of linear ICA, whilst retaining the ability to learn and compensate for the non-linear relationships between time-varying components and their effect on spike waveforms. We go on to demonstrate the efficacy of the model experimentally on both surface and intramuscular EMG, recovering sources with high accuracy on a number of common non-stationary effects. These include, for the first time we are aware of in the literature, recovery of spinal motor neuron activity from non-isometric surface EMG without windowing.

5.2 Method

5.2.1 Preliminaries

The data generation process of neurophysiological signal can be modelled as the observed transformed activity of a set of M latents, the spiking neuronal sources, for which we assume independence, i.e:

$$p(\mathbf{s}) = \prod_{j=1}^M p(\mathbf{s}_j) \quad (5.2.1)$$

The source vector sample is transformed by a mixing function $\mathbf{f} : \mathbf{s} \rightarrow \mathbf{x}$ into an observation vector sample $\mathbf{x}$ detected by the electrode array. We can formulate the blind source separation

goal as the unsupervised discovery of $\mathbf{f}^{-1} : \mathbf{x} \rightarrow \mathbf{s}$, i.e. we want to find a separation function we can apply to the observations that recovers the underlying spiking activity.

The distribution of each spiking source $p(s_j)$ can be described by:

$$p(s_j) = (1 - \psi)\delta(s_j - c_{null}) + \psi\delta(s_j - c_{spike}) \quad (5.2.2)$$

where $\delta(\cdot)$ is the Dirac delta function, c_{null} and c_{spike} are arbitrary constants of different value and ψ is the probability of spike activation determined by upstream neural drives and the neurons refractory period. The length T recorded time series vector sampled from $p(\mathbf{s}_j)$ can be described by a scaled binary sequence dominated by c_{null} :

$$\mathbf{s}_j(t) = \begin{cases} c_{spike} & \text{if neuron spikes at } t \\ c_{null} & \text{otherwise} \end{cases} \quad (5.2.3)$$

At each time instant t in the recording there exists a length L transfer vector $\mathbf{h}_{ij}$ pairing an activation of the j -th neuron with the resultant detected duration L action potential at the i -th electrode assuming adequate time support. $\mathbf{h}_{ij}$ can itself be described as the output from a deterministic function $\mathbf{g} : \mathbf{z}_j \rightarrow \mathbf{h}_{ij}$ which takes as inputs the values of a latent random vector $\mathbf{z}_j$ which abstracts concepts like the relative position and alignment of the neuron with respect to the electrode and the makeup of the intervening volume conductor. When M sources have non-zero transfer functions, $\mathbf{x}_i(t)$, the resultant summed contributions to the i -th electrode at t can be described by a convolutive mixing model:

$$\mathbf{x}_i(t) = \sum_{j=1}^M \mathbf{h}_{ij}(t) * \mathbf{s}_j(t) + \boldsymbol{\omega}_i(t) \quad (5.2.4)$$

where $\mathbf{h}_{ij}(t) = \mathbf{g}(\mathbf{z}_j(t))$ and $\boldsymbol{\omega}_i(t)$ is an additive noise vector.

We can further simplify to an instantaneous mixing model by extending the observation, source, transfer and noise vectors at each t with L past values, indicated by a tilde $\tilde{\cdot}$ in the below definitions. Generalised to an array of N electrodes, this gives us our full description of the mixing function $\mathbf{f}$:

$$\tilde{\mathbf{x}}(t) = \mathbf{H}(t)\tilde{\mathbf{s}}(t) + \tilde{\boldsymbol{\omega}}(t) \quad (5.2.5)$$

where $\mathbf{H}(t)$ is a matrix constructed from the $\tilde{\mathbf{h}}^T(t)$ extended transfer vector outputs given $\mathbf{z}(t)$ for each source, each of which are triggered by $\tilde{\mathbf{s}}(t)$, the extended vector of N binary sources at time t , which we present below in block format:

$$\begin{aligned} \tilde{\mathbf{s}}(t) &= [\tilde{s}_1(t), \tilde{s}_2(t), \dots, \tilde{s}_j(t), \dots, \tilde{s}_M(t)]^T \\ \tilde{s}_j(t) &= [s_j(t), s_j(t-1), \dots, s_j(t-L+1)] \end{aligned} \quad (5.2.6)$$

and finally the summed impulse responses are then observed by M electrodes as the signal $\tilde{\mathbf{x}}(t)$, which we again present in block format:

$$\begin{aligned} \tilde{\mathbf{x}}(t) &= [\tilde{x}_1(t), \tilde{x}_2(t), \dots, \tilde{x}_i(t), \dots, \tilde{x}_M(t)]^T \\ \tilde{x}_i(t) &= [x_i(t), x_i(t-1), \dots, x_i(t-L+1)] \end{aligned} \quad (5.2.7)$$

The bulk of source separation algorithms in the neurophysiological literature require that the mixing matrix remains fixed over time as a necessary assumption. Specifically, we assume $\mathbf{z}(0) \approx \mathbf{z}(t) \approx \mathbf{z}(T)$, which implies $\mathbf{H}(0) \approx \mathbf{H}(t) \approx \mathbf{H}(T)$. This often means head-fixed intracortical recordings in animal models, or isometric contractions in spinal motor neuron recordings.

When this assumption is violated, source separation quality is usually poor. Variation in the observed action potential waveform causes incorrect class splitting and merging in second order spike sorting algorithms, whilst in higher order blind source separation algorithms these non-stationarities cause optimisation problems and missed spiking events. In recent years, intracortical spike sorting algorithms have begun to incorporate a preprocessing step in their pipelines to correct for a specific type of nonstationarity due to drift in the plane of the linear arrays commonly used for such recordings[175, 176, 171]. This is possible to both estimate and correct because the actual waveform shape is mostly unchanged, just its spatial positioning in the multichannel recording. For other types of drift and for changes due to volume conductor variance, variation in $\mathbf{H}$ with respect to time becomes extremely difficult to account for, and source separation in such scenarios is as of this time an unresolved problem.

In intracortical neurophysiology, the rows of $\mathbf{H}(t)$ are simply the extended action potential waveforms associated with each neuronal source and observed by the electrode arrays after modification by the intervening volume conductor. In intramuscular and surface electromyography, these waveforms are instead the summed action potentials of a pool of muscle fibers being triggered at a one-to-one correspondence by a single spinal motor neuron. These differences have significant implications for the waveform characteristics, but the described forward model is nevertheless valid for all of these modalities.

5.2.2 Independent Component Analysis

We aim to recover the neuron sources using blind source separation techniques to discover a separation function which inverts the mixing process in equation 5.2.4. We will use an ICA framework, in which we take advantage of the independence assumption specified in equation

5.2.1 to learn the function $\mathbf{w} \approx \mathbf{f}^{-1}$:

$$\hat{\mathbf{s}}(t) = \mathbf{w}(\tilde{\mathbf{x}}(t); \boldsymbol{\theta}) \quad (5.2.8)$$

where $\boldsymbol{\theta}_j$ are the learned parameters of the model, which we minimise the negative log likelihood:

$$\mathcal{L}_I(\boldsymbol{\theta}) = -\log p_{\hat{\mathbf{s}}}(\mathbf{w}(\tilde{\mathbf{x}}; \boldsymbol{\theta})) - \log |\det J(\mathbf{w}(\tilde{\mathbf{x}}(t); \boldsymbol{\theta}))| \quad (5.2.9)$$

which we will proceed to optimise through maximisation of higher order statistical estimators. Here $\det J(\cdot)$ is the determinant of the Jacobian, which ensures the distribution remains normalised. In practice we can drop the Jacobian term by constraining $\mathbf{w}$ to be unit norm.

For clarity of explanation, we will initially aim to use a purely linear form of $\mathbf{w}$, i.e. $\boldsymbol{\theta} = \mathbf{W}$, which reduces equation 5.2.8 to:

$$\hat{\mathbf{s}}(t) = \mathbf{W}\tilde{\mathbf{x}}(t) \quad (5.2.10)$$

and equation 5.2.9 to:

$$\mathcal{L}_I(\mathbf{W}) = -\log p_{\hat{\mathbf{s}}}(\mathbf{W}\tilde{\mathbf{x}}) - \log |\det J(\mathbf{W})| \quad (5.2.11)$$

For $\log p_{\hat{\mathbf{s}}}$ we want an estimator that is maximal when the distributions are independent, which by well known theory is analogous to maximising an estimator of a higher order statistic[29]. Spiking sources are extremely sparse, supergaussian and asymmetric, so the polynomial cumulant estimator $\log p_{\hat{\mathbf{s}}} \approx \text{sign}(\hat{\mathbf{s}})\hat{\mathbf{s}}^{|e|}$ works well, where e is a exponent hyper-parameter and $\text{sign}(\cdot)$ returns the element-wise sign. Assuming a constrained optimisation

and using a projection pursuit routine to extract one source at a time, this gives our final independence loss for the j -th source:

$$\mathcal{L}_I(\mathbf{w}_j) = -E[\text{sign}(\hat{\mathbf{s}}_j)\hat{\mathbf{s}}_j^{|e|}] \quad (5.2.12)$$

where E is the expectation.

5.2.3 Estimating a Nonstationarity

Under non-stationary conditions and given an optimal set of $\boldsymbol{\theta}_j$, the predicted source distribution $p(\hat{\mathbf{s}}_j)$ converges on the true distribution $p(\mathbf{s}_j)$. However, when $\mathbf{z}$ and therefore $\mathbf{H}$ vary with time, the fixed $\mathbf{W}$ will not be able to learn to invert the mixing model, and the spike amplitudes of $\hat{\mathbf{s}}_j$ will become modulated based on the unknown form of $\mathbf{g}$. This leads to either convergence on only parts of the true $p(\mathbf{s}_j)$ or complete optimisation failure.

To measure and correct for this modulation we take advantage of our prior knowledge on the distribution of $p(\mathbf{s}_j)$ as specified in equation 5.2.2. We first relax the deterministic component to give a two-distribution Gaussian mixture model (GMM) fitted to the output time series of predictions on $\hat{\mathbf{s}}_j$:

$$q(\hat{\mathbf{s}}_j) = (1 - \pi)\mathcal{N}(\hat{\mathbf{s}}_j \mid c_{null}; \sigma_{null}^2) + \pi\mathcal{N}(\hat{\mathbf{s}}_j \mid c_{spike}; \sigma_{spike}^2) \quad (5.2.13)$$

where ψ in equation 5.2.2 is modelled by a mixing coefficient π and Gaussian priors of variance σ_{null}^2 and σ_{spike}^2 placed over the null and spike values.

Given an optimal set of $\boldsymbol{\theta}_j$ and assuming a well fitted GMM and no noise term, the only source of variation in the null and spike Gaussian distributions is the unexplained variance

caused by a non-stationary $\mathbf{z}$. To measure this we can simply take the summed estimated variance of the Gaussian distributions σ_{null}^2 and σ_{spike}^2 as our second loss term, giving the loss function:

$$\mathcal{L}_N(\boldsymbol{\theta}_j) = \lambda(\sigma_{null}^2 + \sigma_{spike}^2)^{\frac{1}{2}} \quad (5.2.14)$$

To avoid fitting a GMM to a long time series with many samples, we instead use an efficient peak finding algorithm to subselect likely members of the null and spike Gaussian distributions with which to initialise the model. We then run a small number of iterations of the expectation-maximisation (EM) algorithm to get the final GMM parameters. This data-driven initialisation of the EM algorithm largely attenuates its slow convergence and tendency to oscillation, allowing for fast and stable calculation of the parameters needed for equation 5.2.14.

5.2.4 HarmonICA: A Quasilinear ICA Model

Of course we have not yet given the model the ability to account for variation in $\mathbf{H}$ with time. As the relationship between $\mathbf{z}$ and $\mathbf{H}$ can be highly nonlinear, we would ideally like to replace our linear model and use a universal function estimator such as a deep neural network for the model predicting the j -th source $\mathbf{w}_j$. However, this immediately causes problems by reducing the identifiability of the ICA model, i.e. optimal versions of $\boldsymbol{\theta}_j$ will give different predictions on the same source[177]. We sidestep this problem with the following model construction, which we name Harmonising Independent Component Analysis or HarmonICA.

Given a neural network NN_j parameterised by $\boldsymbol{\phi}_j$, we separate out the matrix multiplication of the final linear layer, $\mathbf{W}_j$. The purpose of NN_j is, given t , to learn the parameters of an affine transformation of $\mathbf{W}_j$ which compensates for the time-varying $\mathbf{H}$. The modified

$\mathbf{W}_j$ is then directly applied to $\tilde{\mathbf{x}}(t)$ to give $\hat{s}_j(t)$:

$$\begin{aligned}\hat{s}_j(t) &= (\mathbf{a} \otimes \mathbf{W}_j) \tilde{\mathbf{x}}(t) + \mathbf{b} \\ \log(\mathbf{a}), \mathbf{b} &= \textit{split}(NN_j(\boldsymbol{\gamma}(t); \boldsymbol{\phi}_j))\end{aligned}\tag{5.2.15}$$

where $\otimes$ is the element-wise sum on the tiled $\mathbf{W}_j$ and $\textit{split}(\cdot)$ divides the outputs of NN_j into bias and log weight. The intention is that $\mathbf{W}_j$ only parameterises an isometric source separation, whilst the affine transform handles the non-stationary component of the signal. Inspired by Neural Radiance Fields[178], we found that using a multivariate sinusoidal positional encoding for t gave effective generalisation performance to a variety of non-stationary modulations:

$$\boldsymbol{\gamma}(t) = (\sin(\alpha\pi t), \sin((\alpha + 1)\pi t), \dots, \sin((\alpha + \beta)\pi t))\tag{5.2.16}$$

During training the model is optimised using an alternating backpropagation methodology on $\mathcal{L}_I(\mathbf{W}_j)$ and then $\mathcal{L}_N(\boldsymbol{\phi}_j)$. By freezing NN_j during the independence loss update, we prevent nonidentifiability problems whilst still allowing the model to learn complex non-stationary relationships.

The full algorithm is specified in Algorithm 1. After a round of optimisation a quality metric is calculated on the predicted j -th source, for which we used a pseudo-silhouette measure. This was found by using a hard assignment on the per-sample membership probabilities after fitting the GMM, then calculating the average distance of each sample to its assigned distribution mean versus the other mean. If this met a threshold of, the source was added to the accepted sources and its contribution to $\tilde{\mathbf{x}}$ estimated and subtracted, a commonly used "peel off" routine used to prevent convergence on a previously found source in projection

pursuit. Estimation used a local spike triggered average routine.

Algorithm 1 Harmonising Independent Component Analysis

Require: $\tilde{\mathbf{x}}$: length T EMG signal with N channels - filtered, extended and whitened

Require: γ : length T sinusoidal positional encoding of time as per equation ??

Ensure: $\hat{\mathbf{s}}$: length T estimated spinal neuron activity of M sources

```

1: while sources in  $\tilde{\mathbf{x}}$  do
2:   Initialize: Randomly initialize  $\mathbf{W}_j$  and  $\sigma_j$ 
3:   Initialize: Initialize elements of  $\mathbf{a}$  to 1 and  $\mathbf{b}$  to 0
4:   while  $\mathbf{W}_j$  and  $\sigma_j$  not converged do
5:     Independence Update
6:      $\mathbf{W}_j^U \leftarrow \mathbf{a} \otimes \mathbf{W}_j + \mathbf{b}$ 
7:      $\hat{\mathbf{s}}_j \leftarrow \mathbf{W}_j^U \tilde{\mathbf{x}}$ 
8:     Calculate gradient of equation 5.2.12 w.r.t.  $\mathbf{W}_j$ 
9:     Update  $\mathbf{W}_j$  with its gradient
10:    Non-stationarity Compensation Update
11:     $\log(\mathbf{a}), \mathbf{b} \leftarrow NN_j(\gamma; \phi_j)$ 
12:     $\mathbf{W}_j^U \leftarrow \mathbf{a} \otimes \mathbf{W}_j + \mathbf{b}$ 
13:     $\hat{\mathbf{s}}_j \leftarrow \mathbf{W}_j^U \tilde{\mathbf{x}}$ 
14:    Fit GMM to  $\hat{\mathbf{s}}_j$  as per equation 5.2.13
15:    Calculate gradient of equation 5.2.14 w.r.t.  $\phi_j$ 
16:    Update  $\phi_j$  with its gradient
17:  end while
18:  Compute quality metric on  $\hat{\mathbf{s}}_j$ 
19:  if quality metric meets threshold then
20:    Estimate effect of  $\hat{\mathbf{s}}_j$  on  $\tilde{\mathbf{x}}$  and peel off
21:    Add  $\hat{\mathbf{s}}_j$  to  $\hat{\mathbf{s}}$ 
22:  end if
23: end while

```

5.2.5 HarmonICA in Practice

For effective operation on neurophysiological time series, a number of beneficial steps were taken to improve the likelihood of convergence on good sources. In the preprocessing phase the signal was first digitally filtered, and then after extension whitened using the ZCA method.

We found that the number of update steps needed for convergence could be greatly reduced if $\mathbf{W}_j$ had begun to converge on a source prior to running the full optimisation routine. This could be achieved by either initialising $\mathbf{W}_j$ with a high amplitude sample from $\tilde{\mathbf{x}}$ [13], or by

simply running a pre-training phase of a number of independence update steps before starting the compensation updates. In practice we used a combination of both. During training, the parameters of the compensation network σ_j were initialised to by draws from a zero mean gaussian with a small standard deviation, which gave low output values of $\log(\mathbf{a})$ and $\mathbf{b}$ prior to the network being updated. This prevented a big “kick” to a good initialisation of $\mathbf{W}_j$ when the non-stationarity compensation updates start. The RMSprop algorithm was used to control the gradient descent routine for both the independence and compensation updates. Figure 5.2.1 gives some intuition to what this process looks like in practice.

After a source was accepted, it was immediately converted to spike timestamps using the same GMM process. Occasionally the peel-off routine would fail to prevent convergence on a previously found source. This would be identified by calculating a rate of agreement statistic between the new timestamp set and the timestamp sets of previous sources. When this met a certain threshold the source would not be added to the accepted sources, but its contribution to the signal would still be estimated and peeled off to prevent reconvergence.

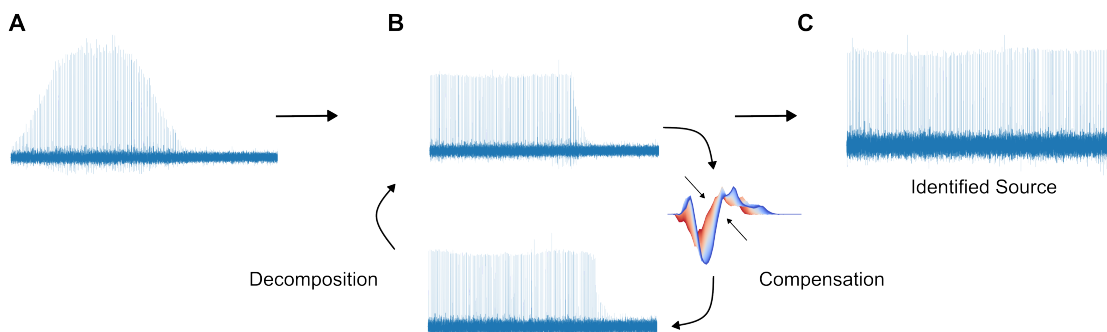


Figure 5.2.1: **a** After an initial pretraining phase, the linear separation vector is converged on a single source. However it lacks the capacity to account for the non-stationary latents in its source prediction. **b** Using alternating backpropagation, HarmonICA cycles between improving the separation vector and training a neural network to adapt the vector such that it accounts for non-stationarities at each time point. **c** At the end of training, the source prediction is no longer modulated by the non-stationary latents.

5.3 Experiments

5.3.1 Real Intramuscular EMG with Synthetic Drift

HarmonICA was tested on the common intramuscular EMG non-stationarity of probe drift using real intramuscular data from a previous study[64]. For this a 10-second recording was taken from a 30% maximum voluntary contraction ankle dorsiflexion dataset sampled at 10 kHz from 32 channels inserted in the Tibialis Anterior muscle of one participant under isometric conditions. Force feedback was shown on a monitor to the participants. The dataset included a manual decomposition of the signal[5], and so had the timestamps of each motor unit active which could be used for a ground truth. Alongside the original static set, sinusoidal channel displacements of amplitude 2mm at periods of 1, 0.5, and 0.25 times the signal duration were embedded into this 32-channel dataset, as well as a superposition of a period 1 drift and period 0.25 drift. The original study was conducted with informed consent from participants, was Institutional Review Board reviewed and conformed to the Declaration of Helsinki.

5.3.2 Simulated Non-isometric Surface EMG

The proposed model was also validated on a simulation of surface EMG recorded under non-isometric conditions[179]. The time-varying volume conductor imposes a complex non-stationarity of the activation potentials in the signal which cannot be estimated using current linear methods. The neuromechanical model uses neurophysiological parameters inferred from a finite element method simulation[65], and musculoskeletal parameters from a forward dynamics simulation[180], integrating both into accurate predictions of how EMG is modu-

lated by dynamic contractions. The output action potential waveforms are convolved with simulated motor neuron activity, whose timestamps also form the ground truth set. The flexor digitorum superficialis was activated at 30% constant neural excitation, resulting in 26 active motor neurons recorded under isometric and dynamic conditions. The generated surface EMG signals corresponded to a 30 second recording from a square 64-channel array placed on top of the muscle at the proximal third of the forearm with 2048 Hz sampling frequency with a 30dB signal-to-noise ratio. In the dynamic condition, non-stationarities in the surface EMG signals were induced by the flexion and extension of the wrist (up to 40° in each direction). These movements followed sinusoids, again with periods equal to 1, 0.5, and 0.25 of the signal length, alongside a superposition of 1 and 0.25.

5.3.3 Real Non-isometric Surface EMG

Real isometric and dynamic recordings from a previous study were also used in validation[181]. Briefly, surface EMG signals from a square 64-channel array were sampled at 2048 Hz from the forearm flexor muscles of one subject. During the experiment, the participant was asked to perform isometric finger presses in a custom dynamometer, with feedback given on screen. Isometric contractions were recorded at 15% of their maximum voluntary contraction at a fixed neutral wrist position over 30s. This finger task was repeated while a motor moved the participant's hand from the neutral position to 30° wrist flexion in a staircase pattern (5s isometric phases, followed by 10s ramps with an incline of 1°/s). The latter induced changes in the muscle length and volume conductor surrounding the sensor arrays. As the movement contained isometric portions of signal, these could be decomposed by linear ICA to give a ground truth in these segments. By carefully extending into the dynamic portion of the

movement, the full time series of the source activity could be reconstructed by calculating the agreement between the timestamps in the overlapping segments. Because it was impossible to tell if a source was not present in a segment because the linear ICA had failed or the unit had genuinely gone silent, only sources which were present through the recording were used as ground truth. As before, the original study was conducted with informed consent from participants, was Institutional Review Board reviewed and conformed to the Declaration of Helsinki.

5.3.4 Implementation Details

During preprocessing, intramuscular EMG was high pass filtered at 100Hz and then extended by 100 past values, whilst surface EMG was bandpassed between 10Hz and 1000Hz before extension by 50 past values.

Pytorch was used for model implementation, with each recording processed on a single RTX6000 GPU for about 20 seconds per source, good or bad, with around 100 source finding attempts. Both RMSprop optimisers used default values except for learning rates of 0.0001 and 0.00001 on intramuscular and surface EMG respectively. Training was terminated when a median interspike interval statistic stopped decreasing. A silhouette threshold of 0.9 was used for both signal types when deciding whether to keep a source. To prevent sources collapsing on a single sample, the exponent e in equation 5.2.12 was set to an initial value of 2 and then incremented to 6 during the independence pretraining phase. Alongside this, the source was clamped during independence updates to be the median value of the top 30 samples.

The non-stationarity compensation neural network architecture was parameterised as a feed-forward neural network with a ReLU activation, batchnorm and dropout between each

hidden layers. There were three such layers, each with a hidden dimension of 64. No activation was placed on the final layer. Positional encoding was with zero phase sine waves with periods set as a fractions of the total length of the time series, we used a 5th, 7th, 9th, 11th and 13th for all studies.

A 1d indexed max pool was used for efficient peak finding for collecting samples with which to fit the GMM. Peaks over a height of 3 standard deviations of the source were initially labelled as spikes, with their average and standard deviation used to initialise those parameters in the spike gaussian distribution in the GMM. Statistics calculated on 1000 other samples were used to initialise the null gaussian distribution. The GMM was then trained by expectation maximisation for 5 steps.

Table 5.1: Performance of HarmonICA, cBSS[14] and ablation on drifted intramuscular EMG

Drift Period (signal lengths)	RoA % (mean $\pm$ std)	Number of Units $> 80\%$ (mean $\pm$ std)	Number of Units $\leq 80\%$ (mean $\pm$ std)
HarmonICA			
None	97.00 $\pm$ 5.90	23.60 $\pm$ 0.55	0.20 $\pm$ 0.45
1	94.60 $\pm$ 11.90	12.60 $\pm$ 1.14	1.20 $\pm$ 0.84
0.5	94.80 $\pm$ 12.50	8.00 $\pm$ 0.00	0.60 $\pm$ 0.55
0.25	88.80 $\pm$ 17.50	6.20 $\pm$ 0.84	1.40 $\pm$ 1.34
1 + 0.25	90.30 $\pm$ 8.90	4.20 $\pm$ 1.79	0.40 $\pm$ 0.89
cBSS[14] (deterministic)			
None	98.80 $\pm$ 18.00	14	0
1	16.40 $\pm$ 12.70	0	18
0.5	14.50 $\pm$ 12.20	0	20
0.25	16.80 $\pm$ 11.60	0	19
1 + 0.25	21.00 $\pm$ 8.38	0	6
HarmonICA with Ablated Compensation			
None	98.70 $\pm$ 1.70	16.20 $\pm$ 1.48	0.00 $\pm$ 0.00
1	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
0.5	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
0.25	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
1 + 0.25	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00

5.4 Results

5.4.1 Intramuscular EMG

HarmonICA was compared to a modern motor unit decomposition algorithm, convolutive blind source separation (cBSS)[14], which is commonly used to extract spinal motor neuron activity from both intramuscular and surface EMG signal. It was also compared to an ablated version of itself, without the compensation module.

HarmonICA outperformed the baseline cBSS in all conditions (see Table 5.1). In the presence of non-stationarities in the signal, cBSS fails to find even a single unit as expected for a purely linear method, whilst the proposed method reliably identifies units, with performance gracefully decreasing in increasingly non-stationary conditions. Notably, this effect is only realized due to the proposed compensation mechanism, as evidenced by the ablation study.

5.4.2 Simulated Surface EMG

Similar trends were observed in the case of simulated surface EMG signals, where HarmonICA outperformed cBSS in all conditions (see Table 5.2).

5.4.3 Experimental Surface EMG

On the fully isometric surface EMG recording, HarmonICA matched the results of cBSS closely, although it consistently found two units across the seeds which were not in the cBSS yield of 29 motor neurons. Over 5 seeds, HarmonICA matched a mean 25.8 of cBSS's units with an average accuracy of 97.31% (std 2.72).

For the two dynamic surface EMG recordings there was no true ground truth, but sources found on small near-isometric portions were carefully curated and then used to reconstruct a

Table 5.2: Performance of HarmonICA, cBSS[14] and ablation on simulated surface EMG

Non-isometry Period (signal lengths)	RoA % (mean $\pm$ std)	Number of Units $> 80\%$ (mean $\pm$ std)	Number of Units $\leq 80\%$ (mean $\pm$ std)
HarmonICA			
None	98.40 $\pm$ 2.66	11.80 $\pm$ 0.45	0.00 $\pm$ 0.00
1	92.22 $\pm$ 1.50	8.67 $\pm$ 0.58	1.33 $\pm$ 1.53
0.50	85.60 $\pm$ 2.74	8.00 $\pm$ 8.00	2.67 $\pm$ 1.53
0.25	91.20 $\pm$ 1.26	8.33 $\pm$ 1.16	1.33 $\pm$ 2.31
1 + 0.25	61.90 $\pm$ 4.01	5.00 $\pm$ 0.00	8.00 $\pm$ 0.00
cBSS[14] (deterministic)			
None	99.60 $\pm$ 0.30	10	0
1	15.00 $\pm$ 17.80	0	9
0.5	19.80 $\pm$ 19.30	0	18
0.25	22.60 $\pm$ 20.50	0	16
1 + 0.25	13.90 $\pm$ 15.00	0	17
HarmonICA with Ablated Compensation			
None	99.49 $\pm$ 0.33	11.60 $\pm$ 0.55	0.00 $\pm$ 0.00
1	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
0.5	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
0.25	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00
1 + 0.25	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00	0.00 $\pm$ 0.00

full time series over the dynamic contraction. Using this method, 6 sources which traversed the entire time series were found in the first recording, and 7 in the second. As with the simulations, and consistent with theory, neither cBSS nor the ablated version of HarmonICA were able to find any units. However, the full version of HarmonICA was able to match 5.2 (std 0.45) of the 6 tracked motor neurons in the first recording, at an average accuracy of 98.66% (std 2.95). In the second recording HarmonICA matched 6.2 (std 0.45) of the 7 units at an accuracy of 96.98% (std 3.66).

5.5 Limitations

The main limitation of this work is the difficulty of validating neurophysiological blind source separation algorithms in real data. Whilst modern full and hybrid simulations have high

fidelity to the real thing, they are ideally only one half of the validation pipeline. We hope in future studies to build a set of source-matched paired surface and intramuscular EMG data during dynamic contractions which would form a gold standard for such studies. We also hope to better understand the how resilient HarmonICA is to different noise conditions. Finally a major limitation of HarmonICA is its computational efficiency, requiring the VRAM and capacities of modern GPUs to build a set of separation vectors and compensation networks. Whilst, by design, the source separation model is more robust to non-stationarity and so will need less recalculation, modifications will be needed to make the pipeline efficient enough to meet interfacing requirements.

5.6 Conclusion

We present a practical approach to the direct decomposition of non-stationary EMG into its constituent motor neuron activity without windowing or *post hoc* correction, including for the first time in the literature the results of direct source separation in surface EMG during a dynamic contraction. HarmonICA has the expressive power to learn the complex non-stationarities caused by common system changes in neurophysiological recordings, whilst retaining the reliability of linear ICA. Our future work will focus on expanding EMG decomposition to more freeform and naturalistic movements, which if successful would form a future control system for motor neuron interfacing systems. We are also interested in seeing how the proposed methodology translates in other neurophysiological domains, such as intracortical recordings.

Chapter 6

Conclusion

6.1 Summary

Electromyography signal is a fascinating construct. On initial observation it appears highly complex, yet unlike other neurophysiological recordings there is tight generative model and a clear set of statistical rules. Moreover, it offers perhaps the most accessible method of extracting neural activity from the human body without a craniotomy. In perfect scenarios, it is striking that motor neuron activity from the spine can be recorded from a few electrodes sitting atop a muscle belly. However, the perfect scenario is usually an engineered one, quite distant from the practical reality. Better algorithms are needed if the technology is to make the jump from the lab to the wild. In this thesis I hope to offer some nascent methods that might help make this possible, taking some of the best methods from modern machine learning and blending them with the established procedures of our field.

In Chapter 2 I demonstrated a pipeline for training deep neural networks to decompose surface EMG into MU activity. At its most simple, it shows that such a transformation is

possible even with a relatively standard recurrent neural network architecture. It also gives us our first “why”; why do we want this seemingly superfluous additional complexity? The GRU network is far more robust to heteroscedasticity than the linear gCKC vectors used for training. Perhaps more intriguingly, when the network was trained on gCKC vectors learned from noisy data, it goes on to outperform those vectors. This implies an ability to outperform a training methodology that provides a noisy label set of timestamps.

One major problem associated with introducing deep learning to neurophysiological time series processing is the additional burden of training a neural network. In Chapter 3 I show that some of this pain can be reduced by pre-training the neural network on the outputs of a finite element method simulation of EMG. This gives the parameters of the architecture an optimal initialisation for quick learning of a decomposition model, greatly improving the reliability of the optimisation.

In Chapter 4 I explored a different learning paradigm, deep metric learning, in which we are interested in embedding discrete sections of a time series into a low dimensional space. I show this is an effective methodology for training MU decomposition networks, with utility beyond just identifying and source separating MUAPs. By introducing a new sampling methodology, locality sensitive mining, I demonstrate that the embedding space can be made much more sensitive to in-class semantic similarity, enabling false positives in a timestamp label set to be identified and removed. In this way we can use the DML embedding to “clean” the output timestamp set from a linear BSS algorithm, whilst simultaneously internalising it for operation on new data.

Finally in Chapter 5 I turned to the problem of decomposing EMG contaminated by non-stationary changes from electrode drift and dynamic contraction. I demonstrate a new

algorithm that closely emulates linear ICA whilst retaining the flexibility of deep neural networks for learning to compensate for non-stationary effects. Using this approach, I show for the first time a direct decomposition of dynamic sEMG, forgoing the need for a local tracking approach.

6.2 Future Perspectives

As development in artificial intelligence moves forward, established methods of neurophysiological source separation will be displaced. However, there remains a gap between what is possible in theory and what is stable in practice. Classical linear methods are likely to remain the dominant paradigm in the near future because they are fast and predictable. They have established theory and decades of progressive methodological improvement. Despite this, they are fundamentally limited, struggling to manage in-band noise, complex superposition and volume conductor variation. Recent years have seen the first green shoots of replacement methodologies, and the field now feels ripe for a revolution. A strong analogy is the domain of computer vision, in the years leading up to the release of AlexNet[182], where deep learning models slowly began to outperform classical methods until suddenly they replaced them entirely once issues of dependability and computational complexity were solved[183].

The major outstanding issue is how we reliably train non-linear functions to solve blind source separation problems. Even here, recent years have seen the perspective of the field move from the view that it was practically impossible to one of cautious optimism as algorithms and theory develop[184]. The key insight is that with appropriate conditioning the model can be constrained such that it becomes identifiable, i.e. it has some guarantee of convergence on the true joint distribution between observations and latents[185]. This

is particularly relevant to EMG, which has the major advantage in having a linear mixing model at every time instant, even if the parameters of that mixing model vary considerably over a recording. The effect of this is to make signals like sEMG recorded during dynamic contractions an ideal test-bed for real world applications of non-linear ICA, which is exciting from both a theoretical and applied perspective.

There are also many improvements that could be made to the current pipeline of linear blind source separation. Some of these are gains in computational efficiency by taking advantage of modern GPUs, for example full ICA rather than projection pursuit is now a more plausible proposition. Full ICA would have significant advantages over current methods, which place a high burden on source quality metrics like the PNR or silhouette to prevent the acceptance of a bad source propagating through the rest of the projection pursuit iterations. Another target for improvement is whitening, which currently invokes a significant computational burden that is problematic for online decomposition methods[186]. One promising avenue in this regard is to use the whitening properties of random projections[187].

Another interesting development is the rise of the foundational model[188], huge neural networks that have been trained on vast quantities of data to be extremely impressive generative models. Whilst finite element method simulations of sEMG and iEMG generation have improved dramatically[189], they still struggle to simulate complex forward models such as dynamic contractions. Some deep generative models have been developed in this domain recently[190, 179], which show potential as entirely *in silico* pipelines of EMG simulation, but an alternative approach would be to train a foundational model on a very large corpus of data. Such a model could be conditioned on kinematics and so generate high quality EMG which could dramatically speed up the development of decomposition algorithms.

Another domain that needs to improve if MU decomposition algorithms are to develop further is the experimental paradigm of two source validation[33, 32]. Current methods are extremely inefficient at creating datasets with a high number of matches between iEMG and sEMG activity. This was more a function of the low yields of wire and needle-based electromyography. However, in recent years there has been an expansion in high density iEMG probes, which has greatly expanded the MU yield[166]. Combined with better ultrasonography techniques that allow close proximity between iEMG and sEMG electrodes, paired yields have dramatically increased. This development should reduce the experimental load needed to validate new MU decomposition algorithms, particularly in the decomposition of dynamic contractions.

6.3 Final Comments

The human nervous system is an endless opera in another room, from which we catch the occasional aria or intermezzo floating through the wall. As the pace of technological progress accelerates, better neurophysiological time series processing will undoubtedly prove foundational in allowing us to explore and eventually exploit our internal information environments, to sit in the front row and understand the motifs and melodies intimately. Until then, I hope the methods presented in this thesis can perhaps make that distant music just slightly clearer.

Bibliography

- [1] S. Andreassen and A. Rosenfalck, “Regulation of the firing pattern of single motor units,” *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 43, no. 10, pp. 897–906, 1980.
- [2] A. Datta and J. Stephens, “Synchronization of motor unit activity during voluntary contraction in man,” *The Journal of physiology*, vol. 422, no. 1, pp. 397–419, 1990.
- [3] C. J. De Luca and Z. Erim, “Common drive of motor units in regulation of muscle force,” *Trends in neurosciences*, vol. 17, no. 7, pp. 299–305, 1994.
- [4] D. Farina, R. Merletti, and D. Stegeman, “Biophysics of the generation of emg signals,” *Electromyography: physiology, engineering, and noninvasive applications*, pp. 81–105, 2004.
- [5] K. C. McGill, Z. C. Lateva, and H. R. Marateb, “EMGLAB: An interactive EMG decomposition program,” *Journal of Neuroscience Methods*, vol. 149, pp. 121–133, Dec. 2005.
- [6] J. R. Florestal, P. A. Mathieu, and A. Malanda, “Automated decomposition of intramuscular electromyographic signals,” *IEEE Transactions on Biomedical Engineering*,

vol. 53, no. 5, pp. 832–839, 2006.

- [7] H. Nakamura, M. Yoshida, M. Kotani, K. Akazawa, and T. Moritani, “The application of independent component analysis to the multi-channel surface electromyographic signals for separation of motor unit action potential trains: Part i—measuring techniques,” *Journal of Electromyography and Kinesiology*, vol. 14, no. 4, pp. 423–432, 2004.
- [8] G. R. Naik, D. K. Kumar, S. P. Arjunan, M. Palaniswami, and R. Begg, “Limitations and applications of ica for surface electromyogram,” in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5739–5742, IEEE, 2006.
- [9] A. Holobar and D. Zazula, “Gradient convolution kernel compensation applied to surface electromyograms,” in *Independent Component Analysis and Signal Separation*, pp. 617–624, Springer Berlin Heidelberg, 2007.
- [10] A. Hyvarinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 10, pp. 626–634, May 1999.
- [11] G. A. Garcia, R. Okuno, and K. Azakawa, “A decomposition algorithm for surface electrode-array electromyogram,” *IEEE engineering in medicine and biology magazine*, vol. 24, no. 4, pp. 63–72, 2005.
- [12] J.-F. Cardoso and A. Souloumiac, “Blind beamforming for non-gaussian signals,” in *IEE proceedings F (radar and signal processing)*, vol. 140, pp. 362–370, IET, 1993.
- [13] A. Holobar and D. Zazula, “Multichannel blind source separation using convolution kernel compensation,” *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4487–4496, 2007.

- [14] F. Negro, S. Muceli, A. M. Castronovo, A. Holobar, and D. Farina, "Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation," *Journal of Neural Engineering*, vol. 13, p. 026027, Feb. 2016.
- [15] C. J. De Luca, "Physiology and mathematics of myoelectric signals," *IEEE Transactions on Biomedical Engineering*, no. 6, pp. 313–325, 1979.
- [16] D. Farina and A. Holobar, "Characterization of human motor units from surface EMG decomposition," *Proceedings of the IEEE*, vol. 104, pp. 353–373, Feb. 2016.
- [17] R. Merletti and S. Muceli, "Tutorial. Surface EMG detection in space and time: Best practices," *Journal of Electromyography and Kinesiology*, vol. 49, p. 102363, 2019.
- [18] A. Del Vecchio, A. Holobar, D. Falla, F. Felici, R. Enoka, and D. Farina, "Tutorial: Analysis of motor unit discharge characteristics from high-density surface emg signals," *Journal of Electromyography and Kinesiology*, vol. 53, p. 102426, 2020.
- [19] M. Besomi, P. W. Hodges, J. Van Dieën, R. G. Carson, E. A. Clancy, C. Disselhorst-Klug, A. Holobar, F. Hug, M. C. Kiernan, M. Lowery, *et al.*, "Consensus for experimental design in electromyography (cede) project: Electrode selection matrix," *Journal of Electromyography and Kinesiology*, vol. 48, pp. 128–144, 2019.
- [20] J. Duchateau and R. M. Enoka, "Human motor unit recordings: origins and insight into the integrated motor system," *Brain research*, vol. 1409, pp. 42–61, 2011.
- [21] D. Farina, F. Negro, S. Muceli, and R. M. Enoka, "Principles of motor unit physiology evolve with advances in technology," *Physiology*, vol. 31, no. 2, pp. 83–94, 2016.

- [22] H. Tankisi, D. Burke, L. Cui, M. De Carvalho, S. Kuwabara, S. D. Nandedkar, S. Rutkove, E. Stålberg, M. J. van Putten, and A. Fuglsang-Frederiksen, “Standards of instrumentation of emg,” *Clinical neurophysiology*, vol. 131, no. 1, pp. 243–258, 2020.
- [23] C. J. De Luca, L. D. Gilmore, M. Kuznetsov, and S. H. Roy, “Filtering the surface emg signal: Movement artifact and baseline noise contamination,” *Journal of biomechanics*, vol. 43, no. 8, pp. 1573–1579, 2010.
- [24] R. Merletti and D. Farina, *Surface electromyography: physiology, engineering, and applications*. John Wiley & Sons, 2016.
- [25] D. Farina, C. Févotte, C. Doncarli, and R. Merletti, “Blind separation of linear instantaneous mixtures of nonstationary surface myoelectric signals,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 9, pp. 1555–1567, 2004.
- [26] R. Merletti and G. Cerone, “Tutorial. surface emg detection, conditioning and pre-processing: Best practices,” *Journal of Electromyography and Kinesiology*, vol. 54, p. 102440, 2020.
- [27] A. Holobar and D. Zazula, “On the selection of the cost function for gradient-based decomposition of surface electromyograms,” in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Aug. 2008.
- [28] M. Chen and P. Zhou, “A novel framework based on FastICA for high density surface EMG decomposition,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, pp. 117–127, Jan. 2016.

- [29] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [30] A. J. Bell and T. J. Sejnowski, “The “independent components” of natural scenes are edge filters,” *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.
- [31] Q. Le, A. Karpenko, J. Ngiam, and A. Ng, “Ica with reconstruction cost for efficient overcomplete feature learning,” *Advances in neural information processing systems*, vol. 24, 2011.
- [32] A. Holobar, M. A. Minetto, and D. Farina, “Accurate identification of motor unit discharge patterns from high-density surface emg and validation with a novel signal-based performance metric,” *Journal of neural engineering*, vol. 11, no. 1, p. 016008, 2014.
- [33] B. Mambrito and C. J. De Luca, “A technique for the detection, decomposition and analysis of the emg signal,” *Electroencephalography and clinical neurophysiology*, vol. 58, no. 2, pp. 175–188, 1984.
- [34] M. Castella and E. Moreau, “A new optimization method for reference-based quadratic contrast functions in a deflation scenario,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3161–3164, IEEE, 2009.
- [35] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, “The expressive power of neural networks: A view from the width,” *Advances in neural information processing systems*, vol. 30, 2017.

- [36] J. Baxter, “A model of inductive bias learning,” *Journal of artificial intelligence research*, vol. 12, pp. 149–198, 2000.
- [37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [38] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, pmlr, 2015.
- [39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [40] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.
- [41] Y. Chauvin and D. E. Rumelhart, *Backpropagation: theory, architectures, and applications*. Psychology press, 2013.
- [42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.
- [43] A. K. Clarke, S. F. Atashzar, A. D. Vecchio, D. Barsakcioglu, S. Muceli, P. Bentley, F. Urh, A. Holobar, and D. Farina, “Deep learning for robust decomposition of high-density surface EMG signals,” *IEEE Transactions on Biomedical Engineering*, vol. 68, pp. 526–534, Feb. 2021.

- [44] A. D. Vecchio, F. Negro, A. Holobar, A. Casolo, J. P. Folland, F. Felici, and D. Farina, “You are as fast as your motor neurons: speed of recruitment and maximal discharge of motor neurons determine the maximal rate of force development in humans,” *The Journal of Physiology*, vol. 597, pp. 2445–2456, Mar. 2019.
- [45] A. Holobar, J. A. Gallego, J. Kranjec, E. Rocon, J. P. Romero, J. Benito-León, J. L. Pons, and V. Glaser, “Motor unit-driven identification of pathological tremor in electroencephalograms,” *Frontiers in Neurology*, vol. 9, Oct. 2018.
- [46] T. Kapelner, I. Vujaklija, N. Jiang, F. Negro, O. C. Aszmann, J. Principe, and D. Farina, “Predicting wrist kinematics from motor unit discharge timings for the control of active prostheses,” *Journal of NeuroEngineering and Rehabilitation*, vol. 16, Apr. 2019.
- [47] C. K. Thompson, F. Negro, M. D. Johnson, M. R. Holmes, L. M. McPherson, R. K. Powers, D. Farina, and C. J. Heckman, “Robust and accurate decoding of motoneuron behaviour and prediction of the resulting force output,” *The Journal of Physiology*, vol. 596, pp. 2643–2659, June 2018.
- [48] A. D. Vecchio and D. Farina, “Interfacing the neural output of the spinal cord: robust and reliable longitudinal identification of motor neurons in humans,” *Journal of Neural Engineering*, Oct. 2019.
- [49] C. J. De Luca, “The use of surface electromyography in biomechanics,” *Journal of applied biomechanics*, vol. 13, no. 2, pp. 135–163, 1997.
- [50] V. Glaser, A. Holobar, and D. Zazula, “Real-time motor unit identification from high-density surface EMG,” *IEEE Transactions on Neural Systems and Rehabilitation En-*

gineering, vol. 21, pp. 949–958, Nov. 2013.

- [51] Y. Zheng and X. Hu, “Real-time isometric finger extension force estimation based on motor unit discharge information,” *Journal of Neural Engineering*, June 2019.
- [52] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.
- [53] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014.
- [54] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, Apr. 2018.
- [55] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.
- [56] A. Samadani, “Gated recurrent neural networks for emg-based hand gesture classification. a comparative study,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1–4, IEEE, 2018.
- [57] M. Simão, P. Neto, and O. Gibaru, “EMG-based online classification of gestures with recurrent neural networks,” *Pattern Recognition Letters*, vol. 128, pp. 45–51, Dec. 2019.
- [58] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *CoRR*, vol. abs/1412.3555, 2014.

- [59] D. Farina, L. Mesin, S. Martina, and R. Merletti, “A surface EMG generation model with multilayer cylindrical description of the volume conductor,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 415–426, Mar. 2004.
- [60] K. G. Keenan, D. Farina, R. Merletti, and R. M. Enoka, “Amplitude cancellation reduces the size of motor unit potentials averaged from the surface EMG,” *Journal of Applied Physiology*, vol. 100, pp. 1928–1937, June 2006.
- [61] R. M. Enoka and A. J. Fuglevand, “Motor unit physiology: Some unresolved issues,” *Muscle & Nerve*, vol. 24, pp. 4–17, Jan. 2001.
- [62] A. Holobar, M. A. Minetto, A. Botter, F. Negro, and D. Farina, “Experimental analysis of accuracy in the identification of motor unit spike trains from high-density surface emg,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 3, pp. 221–229, 2010.
- [63] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, “Techniques of EMG signal analysis: detection, processing, classification and applications,” *Biological Procedures Online*, vol. 8, pp. 11–35, Dec. 2006.
- [64] S. Muceli, W. Poppendieck, F. Negro, K. Yoshida, K. P. Hoffmann, J. E. Butler, S. C. Gandevia, and D. Farina, “Accurate and representative decoding of the neural drive to muscles in humans with multi-channel intramuscular thin-film electrodes,” *The Journal of Physiology*, vol. 593, pp. 3789–3804, Sept. 2015.
- [65] K. Maksymenko, A. K. Clarke, I. Mendez Guerra, S. Deslauriers-Gauthier, and D. Farina, “A myoelectric digital twin for fast and realistic modelling in deep learning,”

Nature Communications, vol. 14, no. 1, p. 1600, 2023.

- [66] D. Farina, I. Vujaklija, R. Brånemark, A. M. Bull, H. Dietl, B. Graimann, L. J. Hargrove, K. P. Hoffmann, H. H. Huang, T. Ingvarsson, H. B. Janusson, K. Kristjánsson, T. Kuiken, S. Micera, T. Stieglitz, A. Sturma, D. Tyler, R. F. Weir, and O. C. Aszmann, “Toward higher-performance bionic limbs for wider clinical use,” *Nature Biomedical Engineering*, 2021.
- [67] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O. C. Aszmann, “The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797–809, 2014.
- [68] D. Farina, I. Vujaklija, M. Sartori, T. Kapelner, F. Negro, N. Jiang, K. Bergmeister, A. Andalib, J. Principe, and O. C. Aszmann, “Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation,” *Nature Biomedical Engineering*, vol. 1, Feb. 2017.
- [69] K. Z. Zhuang, N. Sommer, V. Mendez, S. Aryan, E. Formento, E. D’Anna, F. Artoni, F. Petrini, G. Granata, G. Cannaviello, W. Raffoul, A. Billard, and S. Micera, “Shared human–robot proportional control of a dexterous myoelectric prosthesis,” *Nature Machine Intelligence*, vol. 1, pp. 400–411, 2019.
- [70] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, “Gesture recognition by instantaneous surface EMG images,” *Scientific Reports*, vol. 6, p. 36571, Nov. 2016.

- [71] W. Guo, C. Ma, Z. Wang, H. Zhang, D. Farina, N. Jiang, and C. Lin, “Long exposure convolutional memory network for accurate estimation of finger kinematics from surface electromyographic signals,” *Journal of Neural Engineering*, vol. 18, p. 026027, mar 2021.
- [72] I. Mendez Guerra, D. Y. Barsakcioglu, I. Vujaklija, D. Z. Wetmore, and D. Farina, “Far-field electric potentials provide access to the output from the spinal cord from wrist-mounted sensors,” *Journal of Neural Engineering*, vol. 19, p. 026031, apr 2022.
- [73] A. Holobar and D. Farina, “Noninvasive neural interfacing with wearable muscle sensors: Combining convolutive blind source separation methods and deep learning techniques for neural decoding,” *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 103–118, 2021.
- [74] J. J. Bird, M. Pritchard, A. Fratini, A. Ekart, and D. R. Faria, “Synthetic biological signals machine-generated by GPT-2 improve the classification of EEG and EMG through data augmentation,” *IEEE Robotics and Automation Letters*, vol. 6, pp. 3498–3504, 2021.
- [75] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, “Data augmentation of surface electromyography for hand gesture recognition,” *Sensors (Switzerland)*, vol. 20, p. 4892, 2020.
- [76] F. Wang, S.-h. Zhong, J. Peng, J. Jiang, and Y. Liu, “Data augmentation for EEG-based emotion recognition with deep convolutional neural networks,” in *MultiMedia Modeling* (K. Schoeffmann, T. H. Chalidabhongse, C. W. Ngo, S. Aramvith, N. E. O’Connor, Y.-S. Ho, M. Gabbouj, and A. Elgammal, eds.), (Cham), pp. 82–93, Springer International Publishing, 2018.

- [77] R. A. Zanini and E. L. Colombini, “Parkinson’s disease EMG data augmentation and simulation with DCGANs and style transfer,” *Sensors (Switzerland)*, vol. 20, p. 2605, 2020.
- [78] Y. Wen, S. Avrillon, J. C. Hernandez-Pavon, S. J. Kim, F. Hug, and J. L. Pons, “A convolutional neural network to identify motor units from high-density surface electromyography signals in real time,” *Journal of Neural Engineering*, Mar. 2021.
- [79] T. H. J. M. Gootzen, D. F. Stegeman, and A. van Oosterom, “Finite limb dimensions and finite muscle length in a model for the generation of electromyographic signals,” *Electroencephalography and Clinical Neurophysiology/ Evoked Potentials*, vol. 81, pp. 152–162, 1991.
- [80] A. J. Fuglevand, D. A. Winter, A. E. Patla, and D. Stashuk, “Detection of motor unit action potentials with surface electrodes: influence of electrode size and spacing,” *Biological Cybernetics*, vol. 67, pp. 143–153, 1992.
- [81] D. F. Stegeman and W. H. Linssen, “Muscle fiber action potential changes and surface EMG: A simulation study,” *Journal of Electromyography and Kinesiology*, vol. 2, pp. 130–140, 1992.
- [82] G. Yue, A. J. Fuglevand, M. A. Nordstrom, and R. M. Enoka, “Limitations of the surface electromyography technique for estimating motor unit synchronization,” *Biological Cybernetics*, vol. 73, pp. 223–233, 1995.
- [83] K. Roeleveld, J. H. Blok, D. F. Stegeman, and A. V. Oosterom, “Volume conduction models for surface emg; confrontation with measurements,” *Journal of Electromyogra-*

phy and Kinesiology, vol. 7, pp. 221–232, 1997.

- [84] J. Schneider, J. Silny, and G. Rau, “Influence of tissue inhomogeneities on noninvasive muscle fiber conduction velocity measurements—investigated by physical and numerical modeling,” *IEEE Transactions on Biomedical Engineering*, vol. 38, pp. 851–860, 1991.
- [85] D. P. Botelho, K. Curran, and M. M. Lowery, “Anatomically accurate model of EMG during index finger flexion and abduction derived from diffusion tensor imaging,” *PLoS Computational Biology*, vol. 15, pp. 1–24, 2019.
- [86] V. Carriou, S. Boudaoud, J. Laforet, and F. S. Ayachi, “Fast generation model of high density surface EMG signals in a cylindrical conductor volume,” *Computers in Biology and Medicine*, vol. 74, pp. 54–68, 2016.
- [87] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
- [88] R. Plonsey, “Action potential sources and their volume conductor fields,” *Proceedings of the IEEE*, vol. 65, pp. 601–611, 1977.
- [89] R. Plonsey and D. B. Heppner, “Considerations of quasi-stationarity in electrophysiological systems,” *The Bulletin of mathematical biophysics*, vol. 29, pp. 657–664, December 1967.
- [90] A. Heringa, D. F. Stegeman, G. J. Uijen, and J. P. D. Weerd, “Solution methods of electrical field problems in physiology,” *IEEE Transactions on Biomedical Engineering*, vol. BME-29, pp. 34–42, 1982.

- [91] C. Nicholson and J. A. Freeman, "Theory of current source density analysis and determination of conductivity tensor for anuran cerebellum," *Journal of Neurophysiology*, vol. 38, pp. 356–368, 1975.
- [92] K. H. Pettersen, H. Lindén, A. M. Dale, and G. T. Einevoll, *Extracellular spikes and current-source density*, pp. 92–135. Cambridge, UK: Cambridge University Press, 2010.
- [93] L. A. Peter Knabner, *The Finite Element Method for the Poisson Equation*, pp. 46–91. New York, NY: Springer New York, 2003.
- [94] S. Vallaghé, T. Papadopoulo, and M. Clerc, "The adjoint method for general EEG and MEG sensor-based lead field equations," *Physics in Medicine and Biology*, vol. 54, pp. 135–147, dec 2008.
- [95] R. Merletti and D. Farina, *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd, 2016.
- [96] D. Farina and R. Merletti, "A novel approach for precise simulation of the EMG signal detected by surface electrodes," *IEEE Transactions on Biomedical Engineering*, vol. 48, pp. 637–646, 2001.
- [97] R. Plonsey, "The active fiber in a volume conductor," *IEEE Transactions on Biomedical Engineering*, vol. BME-21, pp. 371–381, 1974.
- [98] P. Rosenfalck, "Intra- and extracellular potential fields of active nerve and muscle fibres. A physico-mathematical analysis of different models," *Acta physiologica Scandinavica. Supplementum*, vol. 321, pp. 1–168, 1969.

- [99] R. E. Burke, D. N. Levine, P. Tsairis, and F. E. Zajac, “Physiological types and histochemical profiles in motor units of the cat gastrocnemius,” *The Journal of Physiology*, vol. 234, pp. 723–748, 1973.
- [100] S. Schiaffino and C. Reggiani, “Fiber types in mammalian skeletal muscles,” *Physiological Reviews*, vol. 91, pp. 1447–1531, 2011.
- [101] L. Modenese and J. Kohout, “Automated generation of three-dimensional complex muscle geometries for use in personalised musculoskeletal models,” *Annals of Biomedical Engineering*, vol. 48, pp. 1793–1804, 2020.
- [102] V. Carriou, J. Laforet, S. Boudaoud, and M. Al Harrach, “Realistic motor unit placement in a cylindrical HD-sEMG generation model,” in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (Orlando, United States), pp. 1704–1707, IEEE, Aug. 2016.
- [103] J. Baxter, “A Bayesian information theoretic model of learning to learn via multiple task sampling,” *Machine learning*, vol. 28, no. 1, pp. 7–39, 1997.
- [104] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.
- [105] G. V. Dimitrov and N. A. Dimitrova, “Precise and fast calculation of the motor unit potentials detected by a point and rectangular plate electrode,” *Medical Engineering*

and Physics, vol. 20, pp. 374–381, 1998.

- [106] D. Farina, L. Mesin, S. Martina, and R. Merletti, “A surface EMG generation model with multilayer cylindrical description of the volume conductor,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 415–426, 2004.
- [107] A. K. Clarke and D. Farina, “Deep metric learning with locality sensitive mining for self-correcting source separation of neural spiking signals,” *IEEE Transactions on Cybernetics*, 2023.
- [108] E. Stark and M. Abeles, “Predicting movement from multiunit activity,” *Journal of Neuroscience*, vol. 27, pp. 8387–8394, Aug. 2007.
- [109] E. Drebitz, B. Schledde, A. K. Kreiter, and D. Wegener, “Optimizing the yield of multi-unit activity by including the entire spiking activity,” *Frontiers in Neuroscience*, vol. 13, Feb. 2019.
- [110] W. Simon, “The real-time sorting of neuro-electric action potentials in multiple unit studies,” *Electroencephalography and Clinical Neurophysiology*, vol. 18, pp. 192–195, Feb. 1965.
- [111] W. H. Calvin, “Some simple spike separation techniques for simultaneously recorded neurons,” *Electroencephalography and Clinical Neurophysiology*, vol. 34, pp. 94–96, Jan. 1973.
- [112] H. G. Rey, C. Pedreira, and R. Q. Quiroga, “Past, present and future of spike sorting techniques,” *Brain Research Bulletin*, vol. 119, pp. 106–117, Oct. 2015.

- [113] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın, M. Barbic, T. J. Blanche, V. Bonin, J. Couto, B. Dutta, S. L. Gratiy, D. A. Gutnisky, M. Häusser, B. Karsh, P. Ledochowitsch, C. M. Lopez, C. Mitelut, S. Musa, M. Okun, M. Pachitariu, J. Putzeys, P. D. Rich, C. Rossant, W. lung Sun, K. Svoboda, M. Carandini, K. D. Harris, C. Koch, J. O’Keefe, and T. D. Harris, “Fully integrated silicon probes for high-density recording of neural activity,” *Nature*, vol. 551, pp. 232–236, Nov. 2017.
- [114] S. Muceli, W. Poppendieck, K.-P. Hoffmann, S. Dosen, J. Benito-León, F. O. Barroso, J. L. Pons, and D. Farina, “A thin-film multichannel electrode for muscle recording and stimulation in neuroprosthetics applications,” *Journal of Neural Engineering*, vol. 16, p. 026035, Feb. 2019.
- [115] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and H. K. D., “Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels,” *bioRxiv preprint*, June 2016.
- [116] J. E. Downey, N. Schwed, S. M. Chase, A. B. Schwartz, and J. L. Collinger, “Intracortical recording stability in human brain–computer interface users,” *Journal of neural engineering*, vol. 15, no. 4, p. 046016, 2018.
- [117] C. Chen, W. Guo, C. Ma, Y. Yang, Z. Wang, and C. Lin, “semg-based continuous estimation of finger kinematics via large-scale temporal convolutional network,” *Applied Sciences*, vol. 11, no. 10, p. 4678, 2021.
- [118] C. Rossant, S. N. Kadir, D. F. Goodman, J. Schulman, M. L. Hunter, A. B. Saleem, A. Grosmark, M. Belluscio, G. H. Denfield, A. S. Ecker, *et al.*, “Spike sorting for large,

dense electrode arrays,” *Nature neuroscience*, vol. 19, no. 4, pp. 634–641, 2016.

- [119] S. Pancholi, A. Giri, A. Jain, L. Kumar, and S. Roy, “Source aware deep learning framework for hand kinematic reconstruction using eeg signal,” *IEEE Transactions on Cybernetics*, 2022.
- [120] H. Lorach, A. Galvez, V. Spagnolo, F. Martel, S. Karakas, N. Interling, M. Vat, O. Faivre, C. Harte, S. Komi, *et al.*, “Walking naturally after spinal cord injury using a brain–spine interface,” *Nature*, pp. 1–8, 2023.
- [121] J.-H. Jeong, J.-H. Cho, B.-H. Lee, and S.-W. Lee, “Real-time deep neurolinguistic learning enhances noninvasive neural language decoding for brain–machine interaction,” *IEEE Transactions on Cybernetics*, 2022.
- [122] R. I. Kumar, M. M. Mallette, S. S. Cheung, D. W. Stashuk, and D. A. Gabriel, “A method for editing motor unit potential trains obtained by decomposition of surface electromyographic signals,” *Journal of Electromyography and Kinesiology*, vol. 50, p. 102383, Feb. 2020.
- [123] F. Hug, S. Avrillon, A. Del Vecchio, A. Casolo, J. Ibanez, S. Nuccio, J. Rossato, A. Holobar, and D. Farina, “Analysis of motor unit spike trains estimated from high-density surface electromyography is highly reliable across operators,” *Journal of Electromyography and Kinesiology*, vol. 58, p. 102548, 2021.
- [124] D. Carlson and L. Carin, “Continuing progress of spike sorting in the era of big data,” *Current Opinion in Neurobiology*, vol. 55, pp. 90–96, Apr. 2019.

- [125] P. Yger, G. L. Spampinato, E. Esposito, B. Lefebvre, S. Deny, C. Gardella, M. Stimberg, F. Jetter, G. Zeck, S. Picaud, J. Duebel, and O. Marre, “A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo,” *eLife*, vol. 7, Mar. 2018.
- [126] Y. Wen, S. J. Kim, S. Avrillon, J. T. Levine, F. Hug, and J. L. Pons, “A deep cnn framework for neural drive estimation from hd-emg across contraction intensities and joint angles,” *bioRxiv*, 2022.
- [127] D. F. Nettleton, A. Orriols-Puig, and A. Fornells, “A study of the effect of different types of noise on the precision of supervised learning techniques,” *Artificial intelligence review*, vol. 33, no. 4, pp. 275–306, 2010.
- [128] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274, 2018.
- [129] Kaya and Bilge, “Deep metric learning: A survey,” *Symmetry*, vol. 11, p. 1066, Aug. 2019.
- [130] X. Yang, P. Zhou, and M. Wang, “Person reidentification via structural deep metric learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, pp. 2987–2998, Oct. 2019.
- [131] S. Liao and L. Shao, “Graph sampling based deep metric learning for generalizable person re-identification,” *arXiv preprint arXiv:2104.01546*, 2021.

- [132] A. Zhong, X. Li, D. Wu, H. Ren, K. Kim, Y. Kim, V. Buch, N. Neumark, B. Bizzo, W. Y. Tak, S. Y. Park, Y. R. Lee, M. K. Kang, J. G. Park, B. S. Kim, W. J. Chung, N. Guo, I. Dayan, M. K. Kalra, and Q. Li, “Deep metric learning-based image retrieval system for chest radiograph and its clinical applications in COVID-19,” *Medical Image Analysis*, vol. 70, p. 101993, May 2021.
- [133] J. V. Sundgaard, J. Harte, P. Bray, S. Laugesen, Y. Kamide, C. Tanaka, R. R. Paulsen, and A. N. Christensen, “Deep metric learning for otitis media classification,” *Medical Image Analysis*, vol. 71, p. 102034, July 2021.
- [134] Z. Chai, L. Luo, H. Lin, H. Chen, A. Han, and P.-A. Heng, “Deep semi-supervised metric learning with dual alignment for cervical cancer cell detection,” in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, IEEE, 2022.
- [135] P. Pati, A. Foncubierta-Rodríguez, O. Goksel, and M. Gabrani, “Reducing annotation effort in digital pathology: A co-representation learning framework for classification tasks,” *Medical Image Analysis*, vol. 67, p. 101859, Jan. 2021.
- [136] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *arXiv preprint arXiv:1703.07737*, 2017.
- [137] Y. Suh, B. Han, W. Kim, and K. M. Lee, “Stochastic class-based hard example mining for deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7251–7259, 2019.
- [138] H. Xuan, A. Stylianou, and R. Pless, “Improved embeddings with easy positive triplet mining,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Com-*

puter Vision, pp. 2474–2482, 2020.

- [139] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.
- [140] W. Chen, X. Chen, J. Zhang, and K. Huang, “Beyond triplet loss: A deep quadruplet network for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [141] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, “Deep metric learning with angular loss,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2593–2601, 2017.
- [142] X. Li, L. Yu, C.-W. Fu, M. Fang, and P.-A. Heng, “Revisiting metric learning for few-shot image classification,” *Neurocomputing*, vol. 406, pp. 49–58, 2020.
- [143] Z. Fu, Z. Mao, C. Yan, A.-A. Liu, H. Xie, and Y. Zhang, “Self-supervised synthesis ranking for deep metric learning,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [144] H. Coskun, D. J. Tan, S. Conjeti, N. Navab, and F. Tombari, “Human motion analysis with deep metric learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [145] Z. Che, X. He, K. Xu, and Y. Liu, “Decade: a deep metric learning model for multivariate time series,” in *KDD workshop on mining and learning from time series*, sn, 2017.

- [146] Z. Chen, Y. Liu, J. Zhu, Y. Zhang, Q. Li, R. Jin, and X. He, “Deep multiple metric learning for time series classification,” *IEEE Access*, vol. 9, pp. 17829–17842, 2021.
- [147] D. Cheng, Z. Li, Y. Gong, and D. Zhang, “Fusion of multiple person re-id methods with model and data-aware abilities,” *IEEE Transactions on Cybernetics*, vol. 50, pp. 561–571, Feb. 2020.
- [148] M. Ye and P. C. Yuen, “PurifyNet: A robust person re-identification model with noisy labels,” *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2655–2666, 2020.
- [149] Y. Yuan, W. Chen, Y. Yang, and Z. Wang, “In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 354–355, 2020.
- [150] D. Fu, D. Chen, H. Yang, J. Bao, L. Yuan, L. Zhang, H. Li, F. Wen, and D. Chen, “Large-scale pre-training for person re-identification with noisy labels,” *arXiv preprint arXiv:2203.16533*, 2022.
- [151] F. Xu, B. Ma, H. Chang, and S. Shan, “PRDP: Person reidentification with dirty and poor data,” *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.
- [152] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *European conference on computer vision*, pp. 499–515, Springer, 2016.

- [153] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, “Metric learning with adaptive density discrimination,” *arXiv preprint arXiv:1511.05939*, 2015.
- [154] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, “No fuss distance metric learning using proxies,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 360–368, 2017.
- [155] S. Kim, D. Kim, M. Cho, and S. Kwak, “Proxy anchor loss for deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3238–3247, 2020.
- [156] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, “Softtriple loss: Deep metric learning without triplet sampling,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6450–6458, 2019.
- [157] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou, “Deep variational metric learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 689–704, 2018.
- [158] Y. Liu, X. Gao, J. Han, and L. Shao, “A discriminative cross-aligned variational autoencoder for zero-shot learning,” *IEEE Transactions on Cybernetics*, 2022.
- [159] Y. Kim and W. Park, “Multi-level distance regularization for deep metric learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1827–1835, 2021.
- [160] S. Ji, Z. Zhang, S. Ying, L. Wang, X. Zhao, and Y. Gao, “Kullback-leibler divergence metric learning,” *IEEE Transactions on Cybernetics*, pp. 1–12, 2020.

- [161] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [162] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” *arXiv preprint arXiv:1701.06538*, 2017.
- [163] G. Algan and I. Ulusoy, “Label noise types and their effects on deep learning,” *arXiv preprint arXiv:2003.10471*, 2020.
- [164] Z. Li, Q. Li, P. Huang, H. Xia, and G. Li, “Human-in-the-loop adaptive control of a soft exo-suit with actuator dynamics and ankle impedance adaptation,” *IEEE Transactions on Cybernetics*, 2023.
- [165] M. A. Díaz, M. Voß, A. Dillen, B. Tassignon, L. Flynn, J. Geeroms, R. Meeusen, T. Verstraten, J. Babič, P. Beckerle, *et al.*, “Human-in-the-loop optimization of wearable robotic devices to improve human–robot interaction: A systematic review,” *IEEE Transactions on Cybernetics*, 2022.
- [166] S. Muceli, W. Poppendieck, A. Holobar, S. Gandevia, D. Liebetanz, and D. Farina, “Blind identification of the spinal cord output in humans with high-density electrode arrays implanted in muscles,” *Science advances*, vol. 8, no. 46, p. 5040, 2022.
- [167] CTRL Labs at Reality Labs, D. Sussillo, P. Kaifosh, and T. Reardon, “A generic noninvasive neuromotor interface for human-computer interaction,” *bioRxiv*, pp. 2024–02, 2024.

- [168] T. Kapelner, M. Sartori, F. Negro, and D. Farina, “Neuro-musculoskeletal mapping for man-machine interfacing,” *Scientific Reports*, vol. 10, no. 1, p. 5834, 2020.
- [169] J. Eden, M. Bräcklein, J. Ibáñez, D. Y. Barsakcioglu, G. Di Pino, D. Farina, E. Burdet, and C. Mehring, “Principles of human movement augmentation and the challenges in making it a reality,” *Nature Communications*, vol. 13, no. 1, p. 1345, 2022.
- [170] G. Dominijanni, S. Shokur, G. Salvietti, S. Buehler, E. Palmerini, S. Rossi, F. De Vignemont, A. d’Avella, T. R. Makin, D. Prattichizzo, *et al.*, “The neural resource allocation problem when enhancing human bodies with extra robotic limbs,” *Nature Machine Intelligence*, vol. 3, no. 10, pp. 850–860, 2021.
- [171] M. Pachitariu, S. Sridhar, J. Pennington, and C. Stringer, “Spike sorting with kilosort4,” *Nature Methods*, pp. 1–8, 2024.
- [172] S. Garcia, C. Windolf, J. Boussard, B. Dichter, A. P. Buccino, and P. Yger, “A modular implementation to handle and benchmark drift correction for high-density extracellular recordings,” *Eneuro*, vol. 11, no. 2, 2024.
- [173] C. Chen, S. Ma, X. Sheng, D. Farina, and X. Zhu, “Adaptive real-time identification of motor unit discharges from non-stationary high-density surface electromyographic signals,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 12, pp. 3501–3509, 2020.
- [174] M. Kramberger and A. Holobar, “On the prediction of motor unit filter changes in blind source separation of high-density surface electromyograms during dynamic muscle contractions,” *IEEE Access*, vol. 9, pp. 103533–103540, 2021.

- [175] C. Windolf, H. Yu, A. C. Paulk, D. Meszéna, W. Muñoz, J. Boussard, R. Hardstone, I. Caprara, M. Jamali, Y. Kfir, *et al.*, “Dredge: robust motion correction for high-density extracellular recordings across species,” *BioRxiv*, 2023.
- [176] M. Pachitariu, S. Sridhar, and C. Stringer, “Solving the spike sorting problem with kilosort,” *BioRxiv*, 2023.
- [177] I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen, “Variational autoencoders and nonlinear ica: A unifying framework,” in *International Conference on Artificial Intelligence and Statistics*, pp. 2207–2217, PMLR, 2020.
- [178] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [179] S. Ma, I. Mendez Guerra, A. H. Caillet, J. Zhao, A. K. Clarke, K. Maksymenko, S. Deslauriers-Gauthier, X. Sheng, X. Zhu, and D. Farina, “NeuroMotion: Open-source simulator with neuromechanical and deep network models to generate surface EMG signals during voluntary movement,” *BioRxiv*, 2023.
- [180] D. C. McFarland, B. I. Binder-Markey, J. A. Nichols, S. J. Wohlman, M. De Bruin, and W. M. Murray, “A musculoskeletal model of the hand and wrist capable of simulating functional tasks,” *IEEE Trans. Biomed. Eng.*, vol. 70, no. 5, pp. 1424–1435, 2023.
- [181] I. Mendez Guerra, D. Y. Barsakcioglu, and D. Farina, “Wearable neural interfaces: Real-time identification of motor neuron discharges in dynamic motor tasks,” *BioRxiv*, 2024.

- [182] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [183] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from alexnet: A comprehensive survey on deep learning approaches,” *arXiv preprint arXiv:1803.01164*, 2018.
- [184] A. Hyvärinen, I. Khemakhem, and H. Morioka, “Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning,” *Patterns*, vol. 4, no. 10, 2023.
- [185] I. Khemakhem, R. Monti, D. Kingma, and A. Hyvarinen, “Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12768–12778, 2020.
- [186] D. Y. Barsakcioglu and D. Farina, “A real-time surface emg decomposition system for non-invasive human-machine interfaces,” in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 1–4, IEEE, 2018.
- [187] A. Camuto, M. Willetts, C. Holmes, B. Paige, and S. Roberts, “Learning bijective feature maps for linear ica,” in *International Conference on Artificial Intelligence and Statistics*, pp. 3655–3663, PMLR, 2021.
- [188] M. Awais, M. Naseer, S. Khan, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, and F. S. Khan, “Foundational models defining a new era in vision: A survey and outlook,”

arXiv preprint arXiv:2307.13721, 2023.

- [189] D. Pereira Botelho, K. Curran, and M. M. Lowery, “Anatomically accurate model of EMG during index finger flexion and abduction derived from diffusion tensor imaging,” *PLOS Computational Biology*, vol. 15, no. 8, p. e1007267, 2019.
- [190] S. Ma, A. K. Clarke, K. Maksymenko, S. Deslauriers-Gauthier, X. Sheng, X. Zhu, and D. Farina, “Human biophysics as network weights: Conditional generative models for ultra-fast simulation,” *arXiv*, 2022.

Appendix

Chapter 2 was originally published as *Deep Learning for Robust Decomposition of High-Density Surface EMG Signals* in *IEEE Transactions on Biomedical Engineering* in 2020 under a Creative Commons Attribution License (CC BY).

Chapter 3 was originally published as *A Myoelectric Digital Twin for Fast and Realistic Modelling in Deep Learning in Nature Communications* in 2023 under a Creative Commons Attribution License (CC BY).

Chapter 4 was originally published as *Deep Metric Learning With Locality Sensitive Mining for Self-Correcting Source Separation of Neural Spiking Signals* in *IEEE Transactions on Cybernetics* in 2023 under a Creative Commons Attribution License (CC BY).

Chapter 5 was originally published as *HarmonICA: Neural Nonstationarity Correction and Source Separation for Motor Neuron Interfaces* on *OpenReview* in 2024 under a Creative Commons Attribution License (CC BY-NC-SA 4.0).