

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Mitigating The Effect Of Electrode Displacement for Robust Neural Interfaces With Wearable Muscle Sensors

Joao Binenbojm de Sa Pereira

Supervisor: Dario Farina

January 6, 2026

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in
Computing of Imperial College London and the Diploma of Imperial College London.

Declaration of Originality

I, Joao Binenbojm de Sa Pereira declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. A full list of references is given in the bibliography.

I acknowledge the use of Generative AI (Gemini, developed by Google) for the purpose of editorial support, specifically for feedback on the narrative structure, as well as on the linguistic clarity of the manuscript. All substantive ideas, data collection, analysis, and conclusions presented in this work are entirely my own.

Chapter 3: This chapter is based on the publication "Tackling Electrode Shift In Gesture Recognition With HD-EMG Electrode Subsets" [126], published in Proceedings of ICASSP 2024, IEEE. The author of this thesis, Joao Binenbojm de Sa Pereira extended the *TensorFlow* infrastructure developed by Balint Hodossy to sample HD-EMG electrode subsets. Joao Binenbojm de Sa Pereira also designed the neural architecture used, and developed the experimental protocol to evaluate benefits in classification generalization. Dimitrios Halatsis contributed to the running of the final experiments and the aggregation of the final results. Professor Farina contributed to the design of the experimental protocol and final analysis.

Chapter 4: The author of this thesis, Joao Binenbojm de Sa Pereira, created the concept, developed the simulations and experimental protocol. Michael Alummoottil contributed to implementing the developed deep learning architecture in *PyTorch* and evaluating on test images. Dimitrios Halatsis contributed to running final experiments and aggregating final results. Professor Farina contributed to the design of the experimental protocol and final analysis.

Chapter 5: The author of this thesis, Joao Binenbojm de Sa Pereira developed the simulations, optimization paradigm and experimental protocol. Dimitrios Halatsis contributed to trialling different EMG simulation methods and Emanuele Abbagnano contributed to evaluation metrics and spike extraction from source estimates. Professor Farina contributed to the design of the experimental protocol and final analysis.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial 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.

Abstract

Brain-computer interfaces offer a direct communication pathway between the nervous system and external devices, promising transformative applications in neurorehabilitation and assistive technology. Among non-invasive modalities, surface electromyography has emerged as a prominent solution, leveraging the muscle as a biological amplifier to provide a high-fidelity, accessible read-out of motor intent. However, widespread adoption is hindered by surface electromyography variability across conditions, with particularly severe performance degradation observed in subject-specific models due to electrode displacement across recording sessions. Existing solutions typically rely on prohibitively large-scale data collection or strenuous calibration procedures.

This thesis aimed to develop machine learning approaches that ensure robust intersession generalization without resource-intensive data collection. Initially, spatial data augmentation using high-density surface electromyography channel subsets was evaluated using a publicly available gesture classification dataset. The approach yielded poor final performances, achieving less than 45% accuracy for classification between eight sustained hand gestures. Consequently, the Spatial Adaptation Layer was introduced: an interpretable, input-based domain adaptation approach designed to virtually correct electrode displacement.

Validated on two independent public datasets with surface high-density electromyography recordings during sustained hand gestures, the introduced layer achieved significant performance improvements over state-of-the-art methods. Specifically, it provided a $> 30\%$ accuracy improvement when using a sparse calibration set of only one sustained gesture. To demonstrate its versatility, this framework was extended to the domain of motor unit decomposition. Validated on both phenomenological simulations and experimental data, the proposed adaptation method significantly reduced simulated electrode displacement (median post-calibration displacement of < 0.25 mm) and enabled the tracking of the majority of originally decomposed motor units. Collectively, these contributions establish a versatile, computationally efficient framework for handling electrode displacement, facilitating reliable neural interfaces for longitudinal use.

Acknowledgements

Firstly, I would like to sincerely thank my supervisor, Professor Dario Farina, and the AI4Health CDT (grant number EP/S023283/1) for providing me with the opportunity to contribute to science, and through the many twists and turns along the way, become a capable and independent researcher.

Beyond Dario's foundational support for the entirety of this project, thanks to his calm, understanding nature I have overcome numerous hurdles throughout the duration of this project, all of which have made me the researcher, engineer and person that I have become.

Secondly, I would like to thank my family. Your incredible support and understanding of hectic schedules, deadlines and submissions has allowed me to progress through the many years of this project. Your kind, loving and unconditional support during this period is immeasurable. It is thanks to your wisdom and warmth that I have reached the end of this milestone.

Crucially, I would like to thank Veronica. Your company, your support and your own incredible achievements have given me the energy and motivation needed for this project. Equally important is your patience when I go on rants about the technical details of EMG, biosignals and optimization algorithms. Your support and dedication are especially appreciated given your legal background and consequent lack of any involvement (or understanding) of the theoretical aspects of my research area. While no statistical analysis was conducted, I felt a significant improvement in my PhD experience with you as a part of it.

Importantly, I would also like to thank Dimitrios, Emanuele, Michael, Rita, and all other invaluable colleagues from Dario's group. I am grateful for your direct support, ideas, brainstorming, and for the ways we supported each other when algorithms or experiments do not go according to plan (as they often times did not). In your individual ways, you have all contributed to my growth as a researcher and human being.

Along these lines, I would also like to extend this gratitude to my fellow AI4Health cohort. Your own cross-disciplinary work has significantly inspired my own. I would also like to personally thank Britta, who has provided me with unimaginable administrative (as well as the occasional emotional) support through the duration of this project.

I would also like to thank all my friends for the general emotional support and your invaluable company. Thanks to you, I was consistently revitalized during difficult and stressful moments.

Last but not least, I would like to show my gratitude to Luna, whom I hypothesise to be the best dog in the world. Thank you for your company throughout the majority of the writing of this thesis and algorithmic development. Your presence provided me with the energy and warmth needed throughout the most challenging periods of this ordeal.

Dedication

To my parents for their endless support and instilling in me the value of hard work.

To Pedro and Sofia, for keeping my time and heart light even through the toughest times.

To Veronica, for having my back every step of the way.

To Luna, for being the best companion during the long nights of coding.

Contents

List of Acronyms	11
Glossary of Terms	15
1 Introduction	17
1.1 Problem Definition	17
1.2 Aims & Objectives	19
1.3 Thesis Breakdown	20
2 Background	22
2.1 EMG: A Non-Invasive Glimpse Of The Nervous System	22
2.1.1 Voluntary Movement Generation	22
2.1.2 Brain-Machine Interfacing	23
2.1.3 Electromyography: A Window Into The Nervous System	24
2.2 Biophysics Of EMG Generation	25
2.2.1 Motor Unit Action Potential: Signal Generation	25
2.2.2 Volume Conduction	26
2.3 Surface EMG Acquisition	27
2.3.1 Signal Instrumentation & Conditioning	28
2.3.2 Surface EMG Montages	28
2.3.3 Surface EMG Armbands	29
2.3.4 High-Density Surface EMG	30
2.4 Motor Unit Decomposition: Interfacing With The Nervous System	31
2.4.1 Electromyography: A Window Into The Nervous System	31
2.4.2 EMG Data Model	31
2.4.3 Optimal Separation Matrix	34
2.4.4 Blind Source Separation	35
2.4.5 From Motor Unit Activity To Motor Intent	38
2.4.6 Applications in Physiology and Neural Engineering	39
2.5 Motor Intent Decoding: Machine Learning	40
2.5.1 Machine Learning	40
2.5.2 Surface EMG-Based Machine Learning Tasks & Applications	43
2.5.3 Traditional Machine Learning	45

2.5.4	Surface EMG-Based Traditional Machine Learning	48
2.5.5	Gradient-Based Optimization	50
2.5.6	Deep Learning	52
2.5.7	Surface EMG-Based Deep Learning	59
2.6	The Problem: Generalization & Robustness	61
2.6.1	Beyond Universal Convergence: The Taxonomy Of Generalization In sEMG Recognition	61
2.6.2	Subject-Specific Intersession Generalization: The Relevance of Electrode Displacement	63
2.7	Current Solutions For Mitigating The Effects Of Electrode Displacement	65
2.7.1	Session-Invariant Representations	65
2.7.2	Domain Adaptation	67
2.7.3	Motor Unit Tracking	70
2.8	Summary & Research Rationale	72
3	Invariance Learning: Tackling Electrode Displacement With HD-EMG Electrode Subsets	73
3.1	Introduction	73
3.2	Methods	74
3.2.1	Dataset	74
3.2.2	Electrode Subset Sampling	76
3.2.3	Preprocessing & Machine Learning	77
3.3	Experimental Analysis	77
3.3.1	Experiment I: Electrode Displacement Generalization	77
3.3.2	Experiment II: Intersession Performance	78
3.4	Conclusion	80
4	Spatial Adaptation Layer: Interpretable Domain Adaptation For High-Density sEMG	81
4.1	Introduction	81
4.2	Related Work	83
4.3	Methods	84
4.3.1	sEMG Intersession Calibration: Partial Class Domain Adaptation	84
4.3.2	Root Mean Square: A Physiologically Meaningful Feature	87
4.3.3	Spatial Adaptation Layer	88
4.3.4	Baseline Normalization	90
4.3.5	Datasets	92
4.3.6	Processing	93
4.3.7	Classification	93
4.3.8	Implementation Details	94
4.3.9	Performance Metrics	94
4.3.10	Domain Adaptation Methods	94
4.4	Results	95
4.4.1	Experiment I: Simulated Transformations Performance	95

4.4.2	Experiment II: Intersession Performance	99
4.4.3	Experiment III: Ablation Study	100
4.4.4	Experiment IV: Effect of Hybrid Optimization Strategy	100
4.4.5	Experiment V: Interpretability Analysis	100
4.5	Discussion	101
4.6	Conclusion	104
5	Tracking Motor Units Across Sessions: A Filter Adaptation Problem	106
5.1	Introduction	106
5.1.1	Motor Unit Decomposition	106
5.1.2	Motor Unit Tracking & Limitations	106
5.1.3	Motor Unit Tracking As A Domain Adaptation Problem	107
5.2	Methods	108
5.2.1	Data	108
5.2.2	Motor Unit Decomposition ¹	110
5.2.3	Decomposition Refinement	111
5.2.4	Spatial Decomposition Adaptation	112
5.2.5	Simulated Spatial Transformations	116
5.2.6	Performance Evaluation	117
5.3	Results	118
5.3.1	Experiment I: Mitigating Spatial Transformations With Phenomenological Simulations	118
5.3.2	Experiment II: Mitigating Simulated Spatial Transformations For Real Data	120
5.4	Discussion	122
5.4.1	Loss Landscape Properties & Optimization	122
5.4.2	Optimization Strategy and Performance	123
5.4.3	Physical Interpretation of Adaptation	123
5.5	Conclusion	124
6	General Discussion & Conclusions	125
6.1	General Summary	125
6.2	Key Findings	125
6.2.1	Spatial Data Augmentation	125
6.2.2	Spatial Input-Based Domain Adaptation	126
6.3	Implications	127
6.3.1	Leveraging Inductive Biases Of Intersession Variability	127
6.3.2	Input-Based Adaptation: From Intersession To Intrasession	128
6.3.3	Motor Unit Tracking: An Input-Based Adaptation Problem	128
6.4	Future Outlook	128
6.4.1	Algorithmic Extensions	128
6.4.2	Motor Unit Tracking: Experimental & Clinical Validation	129
6.4.3	Decomposition As A Universal Calibration Proxy	129

¹The motor unit decomposition procedure and manual spike editing were performed by Emanuele Abbagnano.

6.4.4	Biophysics-Informed Inverse Modelling	129
Bibliography		130
A	Appendix	142
A.1	Root Mean Square: A Physiologically Meaningful Feature	142
A.1.1	EMG Convolutional Model	142
A.1.2	Mathematical Derivation	142
A.1.3	Key Insights	143

List of Figures

1.1	Thesis structure: a schematic showing the sequence of systematic steps undertaken in this project, linking studies to each other and to the success criteria defined. surface electromyography (sEMG) array and gesture set sub-figures were adapted from [38] and [1] respectively.	20
2.1	During movements (or attempted movements), efferent commands are sent from the central nervous system for force generation at the muscles. Using electroencephalography (EEG) sensors, high-fidelity signals can be extracted from the skin surface, enabling numerous downstream brain-machine interface applications (figure adapted from [104]).	24
2.2	A schematic of the cylindrical model of volume conduction for the line-source model of muscle fibres. Given the assumed geometry, volume conduction effects can be computed analytically (figure adapted from [107]).	27
2.3	A visualization of the linear convolutive mixture data model typically assumed in motor unit decomposition (figure adapted from [48]).	32
2.4	A visual representation of a generic convolutional neural network architecture (figure adapted from [156]).	55
2.5	A visual representation of a generic recurrent neural network architecture (figure adapted from [156]).	57
2.6	Visualization of motor unit action potential (MUAP) distributions of tracked motor units (MUs) from the intersession MU tracking study of Martinez-Valdes et al., (2017). Results were obtained from high-density surface electromyography (HD-sEMG) recordings from the vastus lateralis (VL) and the vastus medialis (VM). As shown, for a fixed electrode placement, MUs maintain relatively consistent MUAP distributions across recording sessions (figure adapted from [100]).	71
3.1	Schematic of channel subset sampling procedure for spatial data augmentation applied to the Capgmyo dataset set-up. Three training conditions are considered: (i) training only on the central channel subset, where one channel per acquisition channel is considered; (ii) training/testing on all (128) channels; (iii) training on all valid channel subsets, where only longitudinal translations are considered. . .	75

3.2	The temporal VGGNet variant architecture. The network consists of a number of TVGG blocks consisting of two temporal convolutions, a maxpool and spatial dropout. Each TVGG block doubles the number of channels, while keeping the temporal dimension constant.	76
4.1	Schematic of Spatial Adaptation Layer (SAL) architecture and usage. SAL parameters are frozen and the chosen model is optimized on session #1. On session #2, given a few examples, the model is frozen and SAL parameters are optimized. SAL is similar to a Spatial Transformer module from [78], with the key difference that rather than modelling affine coefficients A_ϕ as a function of the input, ϕ are treated as learnable parameters.	84
4.2	A compilation of results from Experiment I, investigating the ability of the Spatial Adaptation Layer (SAL) to mitigate the effects of simulated spatial transformations: (a,b) The accuracy and f1-score reported with and without SAL-based domain adaptation (DA) across classifiers, number of calibration gestures and datasets. While the Post-Transform baseline (no adaptation) resulted in low performances, SAL-based DA performances were comparable to intrasession performance (Pre-Transform) across conditions. (c) the relationship between accuracy improvement of SAL-based DA with three calibration gestures relative to the baseline. Across classifiers, a significant linear correlation was observed between accuracy improvements and the spatial transformation magnitude. (d) The average electrode distance before and after adaptation paired per trial, showing a significant reduction of average electrode displacement.	96
4.3	Intersession performances across classifiers, adaptation methods, number of calibration gestures and datasets in Experiment II, with a baseline no adaptation baseline (Post-Transform) averaged across classifiers is displayed in red: (a) For the CSL dataset, which uses HD-sEMG arrays with constant interelectrode distance, while Spatial Adaptation Layer (SAL) based domain adaptation (DA) performed comparably when adapting on all available gestures, it significantly outperformed other methods when adapting on fewer gestures. (b) For the Capgmyo dataset, which uses HD-sEMG arrays with variable interelectrode distance, while SAL-based domain adaptation (DA) outperforms other methods when adapting on fewer gestures, its performance was significantly limited in relation to reported performances for CSL.	98
4.4	Statistical analysis of the relationship between magnitude of the learned Spatial Adaptation Layer (SAL) parameters across recording sessions and accuracies given the outcomes of Experiment II for the CSL dataset: (a) For both classifiers, the only parameter that resulted in a significant positive correlation with accuracy improvements was T_x , representing transversal displacements. (b) Significant positive linear correlations are observed between T_x magnitude and accuracy improvement across classifiers, further highlighting results reported in Figure 4.2.	102

- 5.1 Schematic of the steps to simulate sEMG and apply spatial transformation. **(1)** white Gaussian noise based motor unit action potentials (MUAPs) were sampled from $H \times W \times L$ from $\mathcal{N}(0, 1)$, and lowpassed along every dimension with respective cutoff frequencies. **(2)** Independent spike trains were generated for each of N motor units (MUs), and for every channel, summed the convolution of all spike trains with corresponding lowpassed MUAPs. This resulted in a synthetically generated, regular HD-sEMG grid. In practice, the system was initially oversampled with a scaling factor κ , only downsampling after the appropriate transformations were applied. 108
- 5.2 Pipeline of sub-differentiable spatial adaptation for motor unit (MU) decomposition. Firstly, the linear minimum mean square error (LMMSE) optimal separation matrix was split into two components: the inverse covariance matrix of extended observations, and spike-triggered average (STA) templates for all decomposed MUs. **First recording session:** the STA motor unit action potential (MUAP) templates were estimated for the first recording session. A central sub-region of the transformed MUAPs was cropped, such that no post-transformation out-of-bound electrodes were included. **Second recording session:** the second recording session was cropped in the same manner as the MUAPs from the first session. Subsequently, observations were extended and inverse covariance matrix of was estimated. **Pipeline Output:** The matrix product of the cropped STA templates (session 1) and inverse covariance matrix of extended observations (session 2) was applied as a separation matrix to extended observations from the second recording session, resulting in source estimates. From these, the average source negentropy ($J(\mathbf{x})$), a common blind source separation (BSS) learning objective, was computed. Thus, this pipeline provided a sub-differentiable learning objective that could be used to determine the spatial parameters that were BSS-optimal. 111
- 5.3 Qualitative loss landscape analysis as a function of f_c (all conditions evaluated at $SNR = 10$). Given synthetic sEMG at different f_c values, a translation of ($T_{y_0} = -0.8\text{cm}, T_{x_0} = 0.6\text{cm}$) was applied. Subsequently, the normalized negentropy ($\mathcal{L}_{norm}(\mathbf{x})$) was computed based on on this spatially transformed sEMG using the pipeline in Figure 5.2, considering 400 Spatial Adaptation Layer (SAL) corrective translations from a grid within the range $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$, resulting in the above heatmaps. Even as the spatial bandwidth increases, the blind source separation optimal region remains significantly different from other regions in the landscape, suggesting a single, robust global optimum. 118

5.4	Motor unit (MU) decomposition performance metrics for Experiment I, with simulated HD-sEMG and spatial transformations. The RoA, %MU Tracked, and average electrode distance post-adaptation (d cm) were obtained for different optimization methods, signal-to-noise ratio, and spatial lowpass cutoff frequencies. While performance was usually high for mid-range values of f_c , drops in performance were observed for higher and lower values of f_c , an effect that was exacerbated by lower signal-to-noise ratio. As expected, without adaptation, the number of MUs tracked is approximately 0 across all conditions.	119
5.5	Qualitative loss landscape analysis for all subject/recording session pairs during a 25% MVC contraction. To each HD-sEMG grid, a translation sampled from a random uniform distribution was applied, where $T_{y0}, T_{x0} \in [-1.0, 1.0]$ cm. Subsequently, the negentropy was computed based on this spatially transformed sEMG using the pipeline in Figure 5.2, considering 400 Spatial Adaptation Layer (SAL) corrective translations from a grid within the same range as transformations were sampled from ($T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$), resulting in the above heatmaps. Sampled negentropy estimates were normalized by negentropy prior the spatial perturbation, a practical performance upper bound. In experimental recordings from healthy subjects and decomposed manually edited motor units, the blind source separation optimal region remained distinct as a robust global optimum.	120
5.6	Motor unit (MU) decomposition performance metrics for Experiment II, with virtual electrode displacement being applied to real HD-sEMG and decomposed MUs. The RoA, %MU Tracked, and average electrode displacement post-adaptation, d (cm), were obtained for different optimization methods (Latin Hypercube Sampling initialized gradient descent (LHS-GD) , Latin Hypercube Sampling (LHS) and gradient descent (GD)) and the two contraction levels (25% and 50%) recorded. The No Optimization baseline resulted in %MU tracked ≈ 0 across contraction levels. While LHS-GD exhibited the highest median RoA and proportion of MUs, LHS still exhibited a comparable performance distribution across metrics. The GD method exhibited a visibly larger Interquartile Range (IQR) in performance metrics relative to LHS-GD and LHS . This indicated a high trial-to-trial variability, with the performance distribution spanning almost the entire range, from complete failures (lower quartile near 0.00) to highly successful trials (upper quartile reaching ≈ 0.75).	121

List of Tables

3.1	Intrasession test accuracy across Experiment I settings and feature sets (and temporal VGGNet (TVGGNet)), averaged across subjects in DB_a .	78
3.2	Intersession test accuracy across Experiment II settings and feature sets (and TVGGNet), averaged across subjects in DB_b .	79
4.1	Performance by number of adaptation gestures, classifier, median filtering, and Baseline normalization (BN) method.	100
4.2	Comparison for different optimization strategies and classifiers across adaptation set sizes	101
4.3	T_x only vs all parameters across classifiers and $ \mathcal{Y}_{\text{sub}} $	103

List of Acronyms

α -MN alpha motor neuron. [22](#), [24](#), [38](#)

AdaBatch adaptive batch normalization. [82](#), [99](#)

AP action potential. [22](#), [23](#), [25](#), [52](#)

BMI brain-machine interface. [17](#), [23](#), [24](#), [25](#)

BSS blind source separation. [35](#), [37](#), [38](#), [60](#), [62](#), [71](#), [106](#), [107](#), [108](#), [110](#), [112](#), [113](#), [114](#), [117](#), [122](#), [124](#), [128](#), [129](#)

CKC convolution kernel compensation. [35](#), [36](#), [37](#), [38](#), [116](#)

CNN convolutional neural network. [55](#), [56](#), [57](#), [58](#), [59](#), [60](#), [74](#), [77](#), [82](#), [99](#), [104](#), [126](#)

CNS central nervous system. [17](#), [22](#), [23](#), [39](#), [45](#)

CST cumulative spike train. [39](#)

CV conduction velocity. [26](#), [31](#)

DA domain adaptation. [7](#), [67](#), [68](#), [72](#), [77](#), [82](#), [83](#), [84](#), [86](#), [87](#), [94](#), [95](#), [97](#), [98](#), [99](#), [100](#), [101](#), [103](#), [104](#), [105](#), [107](#), [112](#)

DL deep learning. [52](#), [54](#), [58](#), [59](#), [60](#), [63](#), [68](#)

ECoG electrocorticography. [23](#)

EEG electroencephalography. [6](#), [17](#), [23](#), [24](#), [39](#), [66](#)

ETD enhanced time-domain. [77](#)

fastICA fast independent component analysis. [37](#), [38](#), [52](#)

GD gradient descent. [9](#), [51](#), [115](#), [118](#), [119](#), [120](#), [121](#), [122](#), [123](#)

GPU graphics processing unit. [54](#), [56](#), [59](#)

GRU Gated Recurrent Unit. 58, 59

HD-sEMG high-density surface electromyography. 6, 21, 24, 30, 31, 35, 36, 37, 38, 39, 49, 50, 66, 70, 71, 74, 76, 79, 80, 81, 82, 87, 93, 99, 100, 101, 103, 104, 105, 106, 108, 109, 116, 117, 122, 124, 125, 126, 127, 128, 129

HMI human-machine interface. 17

IAP intracellular action potential. 25, 26

ICA independent component analysis. 37, 50

IED interelectrode distance. 29, 30, 31, 66, 70, 92, 93, 97, 99, 100, 101, 104, 105, 110, 116, 117, 120, 122, 123, 124, 127, 128

iEMG intramuscular electromyography. 27, 31, 35, 37, 38, 106

LDA linear discriminant analysis. 48, 66, 68, 77

LFP local field potential. 23

LHS Latin Hypercube Sampling. 9, 50, 90, 94, 100, 101, 103, 115, 118, 119, 120, 121, 122, 123, 124

LHS-GD Latin Hypercube Sampling initialized gradient descent. 9, 115, 118, 119, 120, 121, 122, 123, 124

LMMSE linear minimum mean square error. 34, 36, 38, 111, 112, 113

LMN lower motor neuron. 22

LSTM Long Short-Term Memory unit. 58, 59, 60, 70

M1 primary motor cortex. 22, 23, 24

ML machine learning. 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51, 52, 54, 55, 56, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 72

MU motor unit. 6, 22, 24, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 44, 50, 52, 60, 61, 62, 63, 70, 71, 72, 87, 106, 107, 108, 109, 110, 111, 113, 114, 115, 116, 117, 122, 123, 124, 125, 126, 127, 128, 129

MUA multi-unit activity. 23

MUAP motor unit action potential. 6, 24, 27, 31, 32, 33, 34, 35, 71, 72, 87, 107, 108, 109, 124, 129

MVC maximum voluntary contraction. 37, 38, 109, 110, 120, 122

NLP natural language processing. 40, 45, 57, 58

NMF non-negative matrix factorization. 46

NMJ neuromuscular junction. 25, 26, 31, 39

NN neural network. 52, 53, 54, 55, 57, 58, 59, 68

PCA principal component analysis. 38, 39, 46, 49, 79, 80

PNS peripheral nervous system. 22

RMS root mean square. 49, 87, 88, 90, 91, 92, 103

RNN recurrent neural network. 57, 58, 59, 60, 82

SAL Spatial Adaptation Layer. 21, 83, 88, 89, 93, 94, 95, 96, 97, 99, 100, 101, 103, 104, 105, 107, 112, 113, 115, 116, 117, 124, 125, 126, 127, 128

SCI spinal cord injury. 24, 39, 72

SDA Spatial Decomposition Adaptation. 112, 114, 115, 117, 124

sEMG surface electromyography. 6, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31, 35, 37, 39, 40, 41, 43, 44, 45, 46, 48, 49, 50, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 76, 77, 80, 81, 82, 83, 84, 85, 86, 87, 88, 90, 93, 94, 95, 99, 101, 104, 106, 125

SFAP single-fibre action potential. 31

SGD stochastic gradient descent. 51, 52, 54

SNR signal-to-noise ratio. 18, 23, 24, 28, 64

SOTA state-of-the-art. 29, 54, 56, 57, 59, 77, 93

STA spike-triggered average. 34, 107, 112, 113, 115

STN spatial transformer network. 83, 88, 94

SUA single-unit activity. 23

SVM support vector machine. 46, 47, 49, 51, 60, 66

TA Tibialis Anterior. 110

TD time-domain. 48, 49, 50, 77

TVGGNet temporal VGGNet. 10, 77, 78, 79, 80

UMN upper motor neuron. 22, 24

ViT vision transformer. 59

VR/AR virtual and augmented reality. [43](#), [62](#)

WGN white Gaussian noise. [108](#), [109](#)

Glossary of Terms

Ablation Study An experimental verification procedure where specific components, layers, or features of a system are systematically removed or disabled to isolate and quantify their individual contribution to overall performance. [100](#)

Classification Loss A mathematical objective function that quantifies the discrepancy between a model's predicted categorical labels (such as classified hand gestures) and the true ground-truth targets, producing the error signal used to update model weights. [83](#)

Concept Drift A phenomenon where the statistical properties of the target variable change over time or across domains. In virtual concept drift, the underlying relationship between features and labels remains stable, but changes in the input distribution degrade a static model's accuracy. [84](#), [107](#)

Gesture Within surface electromyography (sEMG) classification and machine learning literature, "gesture" is standardly used as a macro-level class label for specific configurations of sustained muscle contractions mapping to motor intent, decoupled from macroscopic limb displacement or external force generation. In this thesis, extending this definition to encompass attempted movements or sustained muscle efforts that produce no measurable mechanical force is an intentional methodological choice. Because this work focuses fundamentally on the generated peripheral sEMG signal activity as the primary readout of intent, this framing maintains absolute structural generality across both healthy and motor-impaired populations. [18](#), [39](#), [73](#), [81](#), [107](#), [125](#)

Parameter An internal configuration variable of a machine learning model whose value is learned and optimized directly from the training data (e.g., neural network weights and biases, or spatial adaptation translation coefficients ϕ), distinct from hyperparameters which are set manually before training. [21](#), [40](#), [74](#), [82](#), [107](#), [127](#)

Self-Supervised Learning A subset of unsupervised learning where a framework automatically formulates its own proxy supervisory signals or targets directly from the intrinsic structure of the raw input data to train models, eliminating the need for manual external annotation. [58](#)

Sub-Differentiable A generalization of differential calculus to continuous functions that are not everywhere differentiable due to sharp thresholds or non-smooth boundaries (such as bilinear interpolation sampling limits). It provides a valid set of directional derivatives ("subgradients"), enabling standard gradient descent optimization routines to navigate non-smooth loss landscapes. [51](#), [88](#), [115](#)

Unsupervised Learning A category of machine learning algorithms that discover underlying patterns, structural clusters, or source separations directly from data streams without requiring human-annotated target labels (e.g., maximizing source negentropy in blind source separation to isolate individual motor unit spike trains). [18](#), [37](#), [124](#)

Chapter 1

Introduction

1.1 Problem Definition

Human-machine interfaces (HMI) involve the conversion of human intent into commands that can be processed by computer software. A large portion of HMI are mechanical, such as computer keyboards and mice. Whenever humans interact with the external world, the nervous system generates and transmits high-dimensional commands that constitute intention, often limited by low-dimensional interfaces, such as a joystick. Thus, over recent decades, numerous interface modalities have been developed to extract human intent directly from the dynamics of the nervous system. These are known as neural interfaces, or more specifically brain-machine interfaces (BMI) when neural information is extracted from the central nervous system (CNS).

Through direct interfacing with the CNS, BMI have the potential to replace the need for an intermediary mechanical interface between humans and machines, providing a large number of natural degrees-of-freedom, even for users with motor impairments [17]. In particular, while the CNS contains information about numerous human functions, motor commands are relatively more objective than complex and abstract thoughts and have a high degree of spatial consistency in the motor cortex. Thus, a large number of BMI have been developed to extract motor intent, primarily by measuring the electrical activity in target neuronal populations [17].

However, even within the scope of motor intent, these neural phenomena are encoded in the activity of billions of neurons, most of which are located within the bony enclosure of the cranium, making their extraction a non-trivial problem. While intracortical arrays provide the readout of single-neuron activity for tens to hundreds of neurons with high-temporal resolution, their invasiveness and need for surgical procedures render them impractical for daily use. Furthermore, while electroencephalography (EEG) measures cortical activity non-invasively from the scalp, the volume conduction effects of the skull and scalp result in EEG measurements only capturing the aggregated activity of millions of neurons, providing low spatial resolution.

One promising modality is surface electromyography (sEMG): the non-invasive measurement of electrical activity on the skin surface from muscles, originating from motor neurons. Unlike other non-invasive sensing modalities, sEMG targets the peripheral nervous system, leveraging the mus-

cle as a biological amplifier. Since a single spinal motor neuron innervates hundreds to thousands of muscle fibres, its activation generates a large, aggregated electrical response that is easily detectable from the skin surface. Furthermore, because the nervous system converges billions of cortical neurons onto a much smaller pool of spinal motor neurons, sEMG provides a lower-dimensional, high signal-to-noise ratio (SNR) readout of motor intent [41].

The mapping of sEMG to motor intent enables numerous downstream applications, such as naturalistic interaction with augmented/virtual reality (AR/VR) systems [83] and the intuitive control of prosthetic limbs and exoskeletons by amputees [108]. Crucially, sEMG offers unique capabilities for populations with motor impairments. For instance, the majority of spinal cord injury (SCI) subjects exhibit detectable residual activity from sEMG, even in cases of complete paralysis (no movement) [102]. Similar observations were made based on the residual activity of stroke survivors [145]. Thus, sEMG can provide invaluable information about residual motor commands during neurorehabilitation through biofeedback, as well as be used as an interface to control assistive devices, scenarios where mechanical interfaces provide limited to no information about motor intent.

One of the primary ways to extract motor intent from sEMG is through machine learning, a framework of methods that learns a mapping from sEMG recordings to motor quantities of interest. In supervised machine learning, recordings are acquired jointly with labels (supervision signals), often denoting discrete movements such as hand gestures¹ or continuous measurements such as joint angles and forces [156]. In unsupervised learning, instead of labels, prior knowledge of sEMG biophysics and motor control is leveraged to map sEMG to physically interpretable representations, such as extracting the low-dimensional degrees of freedom underlying movement generation [12]. One prominent unsupervised learning framework applied to sEMG is motor unit decomposition, which uses known statistical properties of neural activity to extract the discharge times of a subset of task-relevant neurons [116].

Numerous machine learning methods demonstrate high degrees of intrasession generalization: high motor intent decoding performance when training and testing algorithms on recordings from the same subject and recording session [39, 1]. However, despite machine learning methods yielding robust predictions of motor intent for a large number of degrees-of-freedom in controlled settings, most methods lack robustness in real-world conditions. Since machine learning provides mappings based on the statistical distribution of sEMG observations, changes in sEMG properties invalidate the original mapping. These algorithms often require time-consuming data collection procedures for every new recording session, limiting their clinical and day-to-day use.

Ideal algorithms should exhibit intersubject generalization: accurate and robust motor intent decoding for any subject and recording session with minimal to no calibration. Anatomical differences across subjects, such as muscle geometries and subcutaneous fat thickness, result in large differences in sEMG properties for the same electrode placement. These differences result in poor performance across subjects when mappings are obtained from sEMG recordings from a small

¹Throughout this thesis, the standard term *gesture* is used as a macro-level class label for configurations of sustained isometric muscle effort rather than physical movement. See the front-matter Glossary for the complete methodological definition.

number of subjects [38, 83]. A recent study has shown remarkable improvement by training a "generic" model on sEMG recordings from thousands of subjects, obtaining substantial performance improvements across multiple motor intent decoding tasks [83]. Despite a leap in inter-subject generalization, the personalization of generic models to individual subjects substantially improved performances, suggesting a gap in intersubject performance still remains.

More importantly, sEMG mappings developed in this study are specific to the hardware used, thus not directly applicable to other sEMG acquisition systems. The acquisition of sEMG recordings from a large population requires a lot of time and resources, especially for patients with motor impairments, limiting the practicality of the large-scale data collection approach within healthcare applications.

A weaker, yet sufficient condition for practical sEMG motor intent decoding is subject-specific intersession generalization: the accurate and robust subject-specific motor intent decoding for a single subject with minimal to no calibration for new recording sessions. This simplification enables the use of motor intent decoders on new recording sessions while bypassing the challenge of anatomical differences. Despite the simplification, subject-specific generalization across recording sessions is still limited by various confounding factors such as contraction level, limb pose, movement technique and electrode displacement [156]. In particular, electrode displacement introduces significant challenges, with a shift of even 1 cm severely degrading motor intent decoding performance [63]. While numerous algorithms have been developed to mitigate intersession changes or electrode shift explicitly, high-performing methods typically depend on an extensive amount of calibration data [38, 44] or use processing methods that cannot be implemented in real-time [1]. Therefore, intersession generalization for practical applications remains limited.

1.2 Aims & Objectives

The aim of this thesis is to develop machine learning approaches to improve the robustness of sEMG motor intent estimation without resource-intensive data collection efforts or impractically strenuous calibration sessions. Since intersubject generalization appears to be highly dependent on large datasets, this work only focuses on subject-specific intersession generalization. Given the large amounts of previous studies and publicly available datasets [1, 38], I select hand-gesture classification as the key task to explore robustness improvements. Given its relevance, this thesis focuses on approaches that explicitly mitigate electrode displacement. These robustness improvements should not be classification specific, but applicable to other sEMG machine learning tasks. Thus, methods that show promise in gesture classification are also investigated in the context of tracking motor units across sessions. Thus, my work can be broken down into three key parts:

- I **Intersession Generalization Method Development:** Investigation of methods that extract displacement invariant models, or alternatively, adapt electrode placement specific models to new placements. Both methods aim to improve intersession generalization.
- II **Intersession Gesture Classification:** Validation of implemented methods using multi-session sEMG recordings from hand gesture classification tasks.

III Longitudinal Motor Unit Tracking: Validation of methods that result in sufficient inter-session generalization improvements in gesture classification to motor unit tracking across sessions.

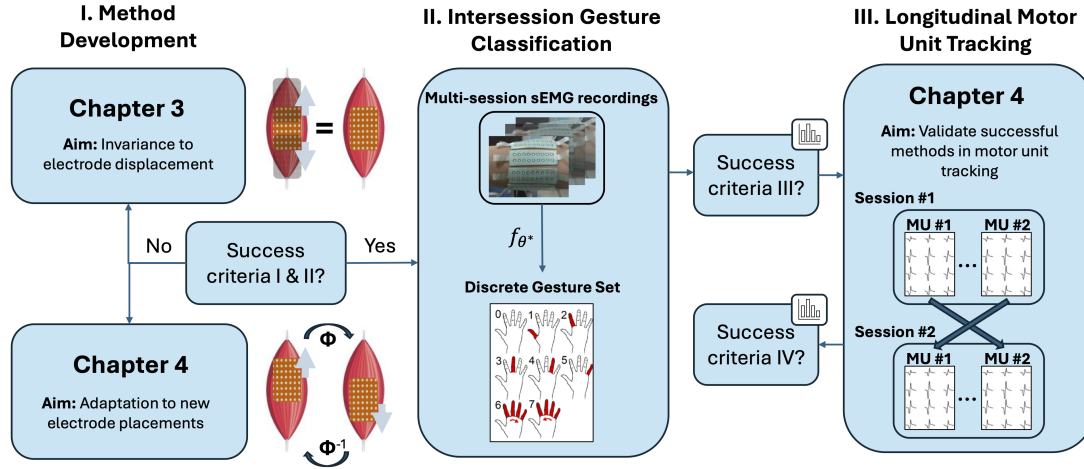


Figure 1.1: Thesis structure: a schematic showing the sequence of systematic steps undertaken in this project, linking studies to each other and to the success criteria defined. sEMG array and gesture set sub-figures were adapted from [38] and [1] respectively.

To measure the success of this line of enquiry, I define the following as **success criteria**:

- I Response time:** As in the canonical work from [39], I define an acceptable delay for windowing/processing to be 300 ms, such that the delay is not perceptible by the user.
- II Gesture Classification Performance:** While difficult to define an accuracy threshold, I defined an acceptable classifier accuracy to be $\geq 85\%$.
- III Number of Calibration Gestures:** While there is no agreement in what constitutes too many calibration gestures (or repetitions), having to collect one repetition of every gesture can be impractical for large gesture sets. Thus, I define the number of allowed calibration gestures for adaptation as three.
- IV Motor Unit Tracking Performance:** Since there is not a substantial body of work on motor unit tracking across sessions, I defined the success criterion as matching or surpassing performances reported in a previous intersession tracking study [100] that did not consider electrode displacement. Thus, the aim is to obtain %MU Tracked to be $\geq 40\%$.

1.3 Thesis Breakdown

This thesis is broken down into 6 chapters, as shown in Figure 1.1. Beyond the introduction of the project motivation and objectives in Chapter 1, Chapter 2 provides a literature review of relevant past work in sEMG, including the neural origins of sEMG generation, acquisition devices, motor unit decomposition methods as well as sEMG-based machine learning. Additionally, Chapter 2 covers past machine learning research that is either relevant or has directly been used in sEMG,

such as classic methods of sEMG classification, novel architectures, and domain adaptation and transfer learning methods, with a special focus on electrode displacement. Popular techniques for electrode displacement mitigation and intersession generalization are also discussed in Chapter 2.

In Chapter 3, a novel method of leveraging high-density surface electromyography (HD-sEMG) as a method of spatial data augmentation for more practical, low-density sEMG is explored as an avenue for generalizing across sessions for subject-specific sEMG classification models. While significant improvements are observed over the baseline, performance is not promising enough to warrant further investigation, acting as an incentive to explore efficient adaptation methods.

In Chapter 4, inspired by Spatial Transformer Networks [78], I introduce the Spatial Adaptation Layer (SAL): an interpretable approach to rapidly find the optimal spatial transformation that mitigates electrode displacement on a uniform sensor array, such as HD-sEMG. With a small number of parameters and entirely preserving the original learned representations, subject-specific intersession performances after adaptation on a single gesture or even three gestures are substantially higher than other popular intersession adaptation methods.

In Chapter 5, the SAL procedure I introduce in Chapter 4 is extended to motor unit decomposition to enable the use of optimal separation vectors from one recording sessions to decompose another session. This is crucial in the context of motor unit tracking: consistent analysis of the same motor units across sessions. This is validated on phenomenologically simulated HD-sEMG, as well as experimental recordings acquired from healthy subjects. The method proposed exhibits strong tracking performance within a range of realistic conditions.

In Chapter 6, I discuss the implications of the most crucial findings in the wider context of HD-sEMG applications, summarize the work carried out, its limitations, and future lines of work.

Chapter 2

Background

2.1 EMG: A Non-Invasive Glimpse Of The Nervous System

2.1.1 Voluntary Movement Generation

The nervous system is the primary communication network of the body, and primarily transmits and receives information with specialized cells known as nerve cells or neurons. When the membrane potential of a neuron, fluctuating about its resting potential, exceeds a certain threshold, an all-or nothing event known as an **action potential (AP)** is generated and propagates through the axon of the neuron. Once the **AP** reaches the synapse, neurotransmitters are released and bind to receptors in the dendrites of receiving neurons, resulting in membrane potential modulations [42]. Given the all-or-nothing property of **APs**, the initiation of an **AP** is commonly referred to as a "firing", "discharge" or "activation". With an interconnected network of billions of neurons, numerous complex commands are simultaneously transmitted. The nervous system is composed of the brain and the spinal cord, jointly known as the **central nervous system (CNS)**, as well as an additional complex network of nerves and ganglia, composed of primarily neurons, known as the **peripheral nervous system (PNS)**.

In movement generation, with the involvement of multiple parts of the brain, a motor command signal is synthesized in and sent from the **primary motor cortex (M1)** through **upper motor neurons (UMNs)**. The **M1** has a somatotopic representation: different parts are associated with the control of different muscles in the body. From the **M1**, **UMNs** transmit motor commands (efferent signals) to interneurons or directly to **lower motor neurons (LMNs)** either in the brainstem (head, neck and face) or the spinal cord (body and limbs). In particular, one sub-type of **LMNs** known as **alpha motor neurons (α -MNs)**, carries these signals to the respective muscles. Each **α -MN** always activates the same subset of muscle fibres, jointly referred to as a **motor unit (MU)**, resulting in muscle contraction. After movement execution, sensory receptors send signals from the **PNS** back to the **CNS** (afferent signals) for movement execution feedback. With different sensing modalities and levels of the nervous system, different proxies of motor intent can be extracted without the need for any dynamics or kinematics measurements.

2.1.2 Brain-Machine Interfacing

Typically, most [brain-machine interfaces \(BMIs\)](#), i.e. systems that translate neural information into computer control commands, have similar working principles. Signals are detected, amplified and filtered. While not exclusively, [BMIs](#) predominantly measure the potential difference between some neurofunctional tissue and a reference to track an aggregation of the dynamics of propagating [APs](#) of the underlying neuronal population. From the detected signals, various algorithms can be used to infer motor intent, which can then be used as control signals for different assistive devices or neurorehabilitation feedback [17].

Direct Brain Interfacing

Given the relevance of the brain in the [CNS](#) in motor planning and command transmission, there are numerous [BMIs](#) that aim to capture the best possible representation of the activity in the [M1](#). Invasive micro-electrodes have historically established the baseline for high-fidelity control, detecting [single-unit activity \(SUA\)](#), [multi-unit activity \(MUA\)](#) or [local field potentials \(LFPs\)](#) [8]. While [SUA](#) provides high resolution activity of single neurons, [MUA](#) provides the activity of a cluster of neurons, still detecting [APs](#) of nearby neurons. [LFPs](#), on the other hand, provide a low-frequency aggregation of a local neural population.

While such micro-electrodes detect highly relevant information for control tasks, its invasive nature (electrodes that pierce through brain tissue) is accompanied by risks and consequently slower research and development. A semi-invasive solution to this is the use of [electrocorticography \(ECoG\)](#) [94], which places a thin film over the surface of the cortex, detecting an aggregation of the activity of the underlying neuronal population. However, despite being less invasive, [ECoG](#) still requires a craniotomy for every re-placement, suffering from similar downsides to the aforementioned techniques.

Therefore, non-invasive [BMIs](#) have been used to detect brain signals from the surface of the scalp, such as [electroencephalography \(EEG\)](#). Without a surgical procedure, these tools provide measures of slow cortical potentials, sensorimotor rhythms and evoked potentials (dynamic responses to specific stimuli) [139]. Instead of measuring changes in electrical activity, [BMIs](#) have also been implemented by measuring blood oxygenation in different tissues, such as for functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS) [13]. By measuring blood oxygenation dynamics, active regions can be tracked during specific periods of time. While [EEG](#) retains good temporal resolution, due to the high electrical impedance of the scalp, signals acquired typically have low [signal-to-noise ratio \(SNR\)](#), only providing measurable signals for activity aggregated over large neuronal populations. While fNIRS and fMRI experience higher spatial resolution, due to the underlying processing, temporal resolution remains low, sometimes referred to as the electromagnetic vs. haemodynamic trade-off, which can limit real-time applications [111].



Figure 2.1: During movements (or attempted movements), efferent commands are sent from the central nervous system for force generation at the muscles. Using EEG sensors, high-fidelity signals can be extracted from the skin surface, enabling numerous downstream brain-machine interface applications (figure adapted from [104]).

2.1.3 Electromyography: A Window Into The Nervous System

As shown in Figure 2.1, efferent motor commands travel from the **M1** through **UMNs** and interneurons to **α -MNs**, which then innervate a fixed group of muscle fibres. When an **α -MN** activates, all innervating muscle fibres in the **MU** obligatorily activate, making the **MU** the smallest functional unit of force generation [57]. Fundamentally, an **MU** is a transducer that converts neural signals into force generation. All the **α -MNs** that innervate the same muscle are known as a motor pool or nucleus [67].

Furthermore, since when an **α -MN** activates, the muscle fibres it innervates also activate, the **MU** amplifies the activity of a single motor neuron by the number of innervating muscle fibres (which ranges from 10-1800 [67]). The aggregated activity from muscle fibres in an **MU** during activation is referred to as a **motor unit action potential (MUAP)**. Hence, in **surface electromyography (sEMG)**, when electrical activity is sensed from the skin surface over the muscle of interest, acquired signals are a naturally amplified, high **SNR** proxy measurement of efferent commands at the spinal cord level. While not widely adopted in clinical applications, a signal processing approach referred to as **MU decomposition** can be applied to **high-density surface electromyography (HD-sEMG)** to extract the specific times **MUs** are active, providing non-invasive measurements of single neuron activity [116]. **MU decomposition** is discussed in more technical detail in Section 2.4.

One promising feature of **BMs**, including **sEMG**, is that while subjects with partial or complete motor impairments are not able to appropriately generate force, residual signals can still be detected at the cortex and periphery. These signals can be acquired during attempted movements and processed to decode motor intent. Residual **sEMG** activity has been detected in up to 64% of clinically motor-complete **spinal cord injury (SCI)** patients [102]. More recently, even for severely affected **SCI** patients, single **MU** activity extracted from **HD-sEMG** was shown to be volitionally

modulated to control a virtual hand on a screen [119]. Additionally, from a sample of stroke patients with complete motor impairments, all subjects exhibited detectable sEMG activity [145]. Thus, sEMG provides a non-invasive, high temporal resolution and fidelity measurement that can be leveraged to decode motor intent and power BMI applications, with detectable activity even in patients unable to generate measurable force.

2.2 Biophysics Of EMG Generation

EMG is a measurement of the electric potential field generated by the aggregation of the depolarization of the outer muscle fibre membrane, known as the sarcolemma. The tissue between the sources of the net electric potential field and the measuring electrodes, referred to as the volume conductor, significantly affects the properties of the detected sEMG [45]. Thus, to analytically understand the properties of EMG, both the signal generation at the sources and the effects of volume conduction must be modelled.

2.2.1 Motor Unit Action Potential: Signal Generation

When a motor neuron activates, its AP is transmitted through the neuromuscular junction (NMJ), generating a local potential gradient in the fibre near the midpoint, corresponding to an inward current density or depolarization zone. Since the depolarization zone is significantly larger than the width of the fibre, its propagation from the NMJ towards the fibre tendons' endings can be modelled by the line source model [45]. The propagating potential is known as the intracellular action potential (IAP), and causes an ionic transmembrane current through the sarcolemma. In this model, the muscle fibre is assumed to be a thin cylinder through which current only flows axially. Given a change in length Δz , the change in potential per unit length can be modelled at the outside and inside of the fibre membrane as follows:

$$\frac{\partial \phi_e}{\partial z} = -I_e r_e, \quad \frac{\partial \phi_i}{\partial z} = -I_i r_i \quad (2.1)$$

where ϕ_e and ϕ_i are the extracellular and intracellular potentials, I_e and I_i are the extracellular and intracellular currents, and r_e and r_i are the extracellular and intracellular resistances per unit length. Furthermore, given the conservation of current, the longitudinal rate of change of the intracellular current is given by:

$$\frac{\partial I_i}{\partial z} = -i_m \quad (2.2)$$

where i_m is the transmembrane current density.

Similarly, the longitudinal rate of change of the extracellular current is determined by the transmembrane current density, as well as the current density due to any indwelling electrodes:

$$\frac{\partial I_e}{\partial z} = i_m + i_p \quad (2.3)$$

To determine the potential gradient, the transmembrane voltage is computed as:

$$V_m = \phi_i - \phi_e \quad (2.4)$$

Combining these relationships, the first and second derivatives of the transmembrane voltage can be determined with respect to longitudinal length:

$$\frac{\partial V_m}{\partial z} = -r_i I_i + r_e I_e \quad (2.5)$$

$$\frac{\partial^2 V_m}{\partial z^2} = (r_e + r_i) i_m + r_e i_p \quad (2.6)$$

Thus, under the line source model, the second derivative of the transmembrane voltage (determined by the propagating **IAP**) is almost proportional to the transmembrane current density. From this relationship, the transmembrane current density source (i_m) can be mathematical derived from the known shape of the (V_m). The **IAP** is typically modelled through various approximations, each consisting of depolarization, repolarization, and a hyperpolarization long afterpotential. Typically, to constrain the generation and extinction of the **IAP**, the integral of the transmembrane current density (i_m) over its longitudinal length is assumed to be zero. For instance, under the triangular **IAP** approximation, its first derivative is a constant value for each phase, and its second derivative (i.e. transmembrane current density) is represented by a series of Dirac delta functions at the change-point of each phase, referred to as a current tripole.

One model of propagation is the progressive appearance of this tripole at the **NMJ** and disappearance at the tendons in both directions [35]. From different models under the same assumptions, it can be shown that, unlike propagating waves in infinitely long wires, EMG signals at different positions are not simply time-delayed versions of each other [45]. In particular, the disappearance at the tendons introduces non-propagating components with different shapes at different locations of the muscle, known as the end-of-fibre effects. **IAP** propagation also depends on the **conduction velocity (CV)**, an inherent property of a muscle fibre that shows a strong dependence on fibre diameter. From different models of **IAP** propagation, the properties of the spatiotemporal distribution of the transmembrane current density can be investigated. However, to understand how these properties relate to **sEMG** signals, volume conduction effects must also be considered.

2.2.2 Volume Conduction

The volume conductor acts as the physical transfer function that maps the transmembrane current density (i_m) derived above to the detected EMG at different locations within the tissue. Since the biological **CV** of **IAPs** is orders of magnitude lower than the speed of light, the assumption of quasi-stationarity can be employed, simplifying volume conduction to an electrostatic problem. In particular, this implies solving the Poisson equations in the electrostatics case, commonly referred to in this context as forward equations [99]. Given the previously modelled transmembrane current source density from a muscle fibre, as well as the conductivity of the surrounding tissue, the

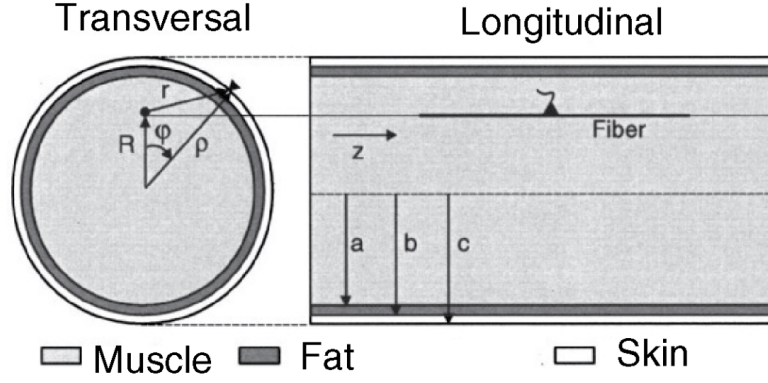


Figure 2.2: A schematic of the cylindrical model of volume conduction for the line-source model of muscle fibres. Given the assumed geometry, volume conduction effects can be computed analytically (figure adapted from [107]).

solution to these equations is the potential at any point in space. For various approximations of the tissue geometry, such as the cylindrical model in Figure 2.2 these equations can be solved analytically [49, 46]. Recent methods have used numerical methods to obtain anatomically realistic EMG distributions from 3D tissue meshes constructed from segmented MRI scans [99].

From these models, it has also been noted that end-of-fibre potentials are considered far-field potentials, decaying more slowly over space than propagating signals. This implies the prominence of end-of-fibre potentials increases in EMG measurements with the distance from the source. This greatly contributes to the problem known as cross-talk in sEMG recordings, where detected signals contain significant contributions from non-propagating components from non-targeted muscles [45]. Through finite volume conductor models, it has also been shown that beyond signal attenuation, the volume conductor acts as a spatial lowpass filter. These effects are dependent on the conductivities of different tissue layers, their thickness, as well as the orientation of the tissue relative to the source. Therefore, each location of skin over a muscle is a mixture of unique spatiotemporal responses to sources deep within the muscle tissue. This highlights the sensitivity of sEMG recordings to electrode displacement, as small deviations from the original location can have large consequences on the volume conduction effects of all active sources.

2.3 Surface EMG Acquisition

As detailed in the previous section, the firing of MUs generates complex electrical potential dynamics that propagate through different layers of tissue. There are two primary levels at which this activity is measured: semi-invasively using needle electrodes, referred to as intramuscular electromyography (iEMG), or non-invasively at the skin surface (sEMG). iEMG recordings provide the gold standard for precisely locating the electrophysiological centre of specific MUs [135] and offer the easiest path to MUAP separability due to their proximity to the source [93]. However, the invasive nature of the implantation requires a complex, highly controlled experimental setup, making long-duration or dynamic monitoring impractical. Therefore, the scope of this thesis is solely on sEMG (wearable muscle sensors).

2.3.1 Signal Instrumentation & Conditioning

In biological tissues, charge is carried by ions in electrolytes and electrons in metal conductors. Therefore, the electrode serves a crucial function: it must be capable of transduction, converting ionic current flow from the tissue into electronic current flow for measurement. This electronic current is then fed into a differential amplifier to obtain a measure of the potential difference relative to another electrode, either from an active location or a reference point. Beyond amplifying the underlying signals, acquisition systems perform analogue signal conditioning, including anti-aliasing filters, followed by digitization with an analogue-to-digital converter (ADC) [105].

For the commonly used wet (or gelled) electrodes, the transduction from ionic to electronic current is mediated by an electrolyte gel. With these, ions from the biological tissue flow into the gel, and through an electrochemical reaction with the electrode metal (often silver or silver chloride), result in a propagated electronic current [105]. Some electrode designs are built to bypass the need for a conductive gel or paste, either by using the body's moisture as the intermediate layer, or by depending on capacitive mechanisms. These latter electrodes are referred to as dry electrodes, and while their placement is significantly more practical and has already been adopted in commercial-grade sEMG devices [83], they often experience significantly higher contact impedance, leading to various downstream complications.

Electrode-skin impedance significantly affects the SNR of the acquired sEMG signals. Firstly, an increase in absolute impedance results in a decrease in the common-mode rejection ratio (CMRR), which can lead to an excessive amount of common-mode noise in the signal. Secondly, higher absolute impedances lead to a greater amount of thermal noise generated. Thus, skin-electrode interface properties are vital to acquiring high-fidelity surface potential measurements. Lastly, impedance mismatch across electrodes can lead to further complications since a differential amplifier is no longer able to perfectly cancel out the noise between them, further decreasing CMRR.

For instance, impedance is inversely proportional to the contact area, suggesting larger electrodes will detect signals with greater SNR. However, not only do larger electrodes add a greater physical constraint on how densely electrodes can be placed, but as long as the electrode-skin impedance is lower than amplifier impedance, the contact surface approximately averages the surface potential [34], introducing spatial lowpassing effects on the acquired signals [49]. Thus, while higher SNR is always desirable, such a property must be balanced with the tolerable amount of spatial lowpassing for the downstream task. Furthermore, since the skin surface has variable properties, skin preparation has significant effects on the measurement impedance. While stripping the surface with tape and rubbing alcohol onto the skin did not lead to significant changes in impedance, rubbing the skin with abrasive paste significantly lowered the impedance relative to all other treatments [130].

2.3.2 Surface EMG Montages

Surface EMG signals are not acquired through single electrodes, but rather as a linear combination of more than one electrode (montages) and a series of differential amplifiers [47]. Such montages can be interpreted as a form of spatial filtering, often used as spatial highpass filters to

counteract the spatial lowpassing effect of volume conduction and increase spatial selectivity of the recording. This framework can be leveraged to minimize the effects of noise present equally across all channels (common-mode noise), such as powerline interference, as well as to minimize the detection of electrical activity from nearby muscles, known as cross-talk.

The most traditional montage is the bipolar (single differential), which measures the difference between two electrodes a short distance (1-3 cm) from each other along the direction of the muscle fibres, mathematically approximating the first spatial gradient of the surface potential [45]. Signals can also be acquired with a monopolar montage, which uses a far-away reference electrode. Bipolar montages were shown to provide greater spatial selectivity, significantly reducing the effect of cross-talk for superficial MUs. However, for deeper MUs, the gradient of surface potential was not large enough to be detected. The advantages and shortcomings of bipolar montages are accentuated in double differential montages, which measure the difference between two adjacent bipolar recordings, providing a measure of the second-order gradient of potential over the skin surface [29]. The trade-off between differential and monopolar montages of spatial selectivity vs. amplitude preservation of deeper MUs in 1D recordings highlights the need for more advanced, 2D acquisition and processing techniques.

2.3.3 Surface EMG Armbands

While single-channel recordings provide a measure of electrical activity of the muscles underneath the sensing region, downstream tasks often require discriminatory information from multiple muscles. Commercial myoelectric control systems (from the domain of prostheses control) traditionally depended on an anatomically-oriented placement: placing sEMG electrodes directly over the bellies of muscles of interest for the downstream task [55]. However, for popular control strategies, no significant difference in performance between an anatomically-oriented placement strategy and an untargeted strategy was detected, where for the latter, the same numbers of electrodes were placed at a constant interelectrode distance (IED) from each other along the circumference of the arm [56]. The apparent match in performance without the need of an expert operator for appropriate placement inspired various future studies that adopted the circumferential, untargeted placement [158, 86, 121].

This untargeted placement strategy likely informed the design of the Myo Armband: an 8-channel, dry-electrode, low-cost consumer-grade sEMG armband [27]. Despite its low sampling rate ($f_s = 200$ Hz) likely introducing aliasing artifacts, myoelectric control performance was not significantly different from the CONV system of bipolar electrodes [103]. Thus, its relatively low-cost, untargeted placement and dry electrodes propagated the use of the Myo Armband in numerous myoelectric control studies [131, 103, 27]. Although the Myo Armband was discontinued in 2018, the initial promise of commercial sEMG systems was arguably a great inspiration behind the sEMG hardware developed for myoelectric control by CTRL-Labs, a company that was acquired by Meta (formerly Facebook) and integrated into Reality Labs in 2019. Since then, with their proprietary sEMG armband, Kaifosh, Reardon, and CTRL-labs at Reality Labs, (2025) achieved state-of-the-art (SOTA) performance for myoelectric controllers that generalize across subjects through the use of deep learning and a dataset of thousands of participants [83]. Further contributions were

made with their two publicly available datasets, each containing sEMG recordings from hundreds of subjects with their sEMG armband for various myoelectric control tasks [138, 143].

While Reality Labs showed the feasibility of developing myoelectric controllers that generalize across subjects, as well as publishing datasets for myoelectric algorithmic development, such control strategies are only applicable to controllers using the same sEMG acquisition set-up as in the originally collected dataset. Furthermore, the scale of the collected dataset (thousands of subjects) is often prohibitively expensive to acquire outside of a commercial setting, as is often the case for medical applications. For single-subject models, the same study reported significantly worse performances across recording sessions, reflecting the lack of robustness of such myoelectric controllers without sufficient data.

2.3.4 High-Density Surface EMG

Traditional sEMG acquisition set-ups focused on acquiring sEMG signals from one or multiple locations with high temporal resolution. However, sEMG over the skin surface is modelled as a three-dimensional signal: two spatial dimensions and one temporal dimension [46]. This implies that low-density electrode placements such as for the aforementioned Myo Armband only capture a small subset of the spatio-temporal dynamics of the underlying signal. A logical progression is to increase the spatial resolution of acquisition set-ups, achieved by decreasing the IED. The acquired signals with this increased spatial sampling density are referred to as HD-sEMG signals. Such high-density systems are typically acquired with two-dimensional uniform arrays (grid with constant IED), providing time-varying images of the spatial distribution of electrical potential [106]. In HD-sEMG images, the value of a "pixel" is the measured sEMG value at a given electrode location.

When digitally sampling a signal, to prevent aliasing, the sampling frequency must satisfy the Nyquist-Shannon sampling theorem, given by $f_s > 2f_{max}$, where f_s is the sampling frequency and f_{max} is the highest frequency component present in the signal. This criterion provides a theoretical guarantee that the sampled signal can be used to reconstruct the underlying analogue (continuous) signal. In other words, the signal is sampled with sufficient density to capture all the information in the signal. The temporal bandwidth of sEMG signals has been studied extensively, usually assumed to be bounded by $f_{max} = 500$ Hz. Hence, common acquisition set-ups capture all the underlying temporal information. Due to the complexity introduced with increased spatial density of electrodes, the Nyquist limit of spatial sampling frequencies has been primarily investigated in a simulation setting. Simulation studies utilizing single fibre potentials have demonstrated that spatial aliasing is avoided when $IED < 1$ cm [106]. Since electrode surfaces have a spatial lowpassing effect on the underlying signals [49], it is likely that IEDs higher than 10 mm can be used reliably.

Given a small enough IED, HD-sEMG images can be safely processed as images, with extensively studied methods from the domain of image processing. This has been used extensively for HD-sEMG imaging to visually interpret the spatiotemporal variations across conditions over a muscle [106]. From these images, image processing techniques can also be used to estimate motor intent for myoelectric control, such as segmentation of active regions [136, 1] and image-based deep

learning [38, 160]. Furthermore, the understanding of the underlying geometry can be leveraged by explicitly accounting for spatial information. For instance, HD-sEMG over the forearm was processed as time-varying images using the Watershed algorithm to determine the location of the ulna, a location in space with low expected EMG activity [1]. The estimated ulna location was used to calibrate myoelectric control algorithms by virtually readjusting the grid in space.

Lastly, when the IED is small, the number of channels in the vicinity of the same MUs is increased, providing different perspectives of the spatiotemporal impulse response of a MU at the skin surface. While for a single channel, the response of different MUs may appear similar, the more channels in the vicinity of the MU, the better the response of each active MU can be discriminated. This was investigated by measuring the normalized mean-square difference between MUAP shapes of different MUs for experimental recordings and simulated EMG [53]. While a single-channel did not provide enough information to reliably discriminate a significant number of MUs, increasing the number of channels and density increased the number of MU pairs that were supra-threshold. With enough discriminability between sources, blind source separation methods were developed and extensively studied that estimate the discharge times and MUAP shapes of underlying sources during isometric contractions [70, 116], a process known as MU decomposition (see Section 2.4 for decomposition information). More recently, larger and denser grids resulted in the decomposition of a more representative pool of MUs [15]. Thus, besides HD-sEMG images, denser recordings can provide a reliable window into the efferent motor commands for the muscle(s) under the recording surface.

2.4 Motor Unit Decomposition: Interfacing With The Nervous System

2.4.1 Electromyography: A Window Into The Nervous System

MU decomposition separates the contributions of individual MU activity from EMG recordings. From the simultaneous activity of MUs innervating a muscle, different properties of neural control of movement can be investigated. Classically, MU activity is extracted with the use of iEMG. However, due to the invasive nature of intramuscular recordings, this approach is limited in practical applications and daily use. Thus, numerous methods have been developed to decompose MU activity from sEMG recordings.

2.4.2 EMG Data Model

sEMG recordings measure the superposition of underlying activations of single-fibres, known as single-fibre action potentials (SFAPs). The activation of a muscle fibre generate a propagating potential from the NMJ towards the tendons. Since single-fibres have the same CV for the same muscle, and can be seen as propagating synchronously under the assumption of sufficiently close NMJ locations. By treating the simultaneous SFAPs as phase-locked, their aggregate response at the surface level can be modelled as the joint response to a single MU discharge, referred to as a MUAP [71]. Thus, sEMG is often mathematically represented as a linear convolutive mixture of sparse sources, as visualized in Figure 2.3. This comes from the assumption that during stationary

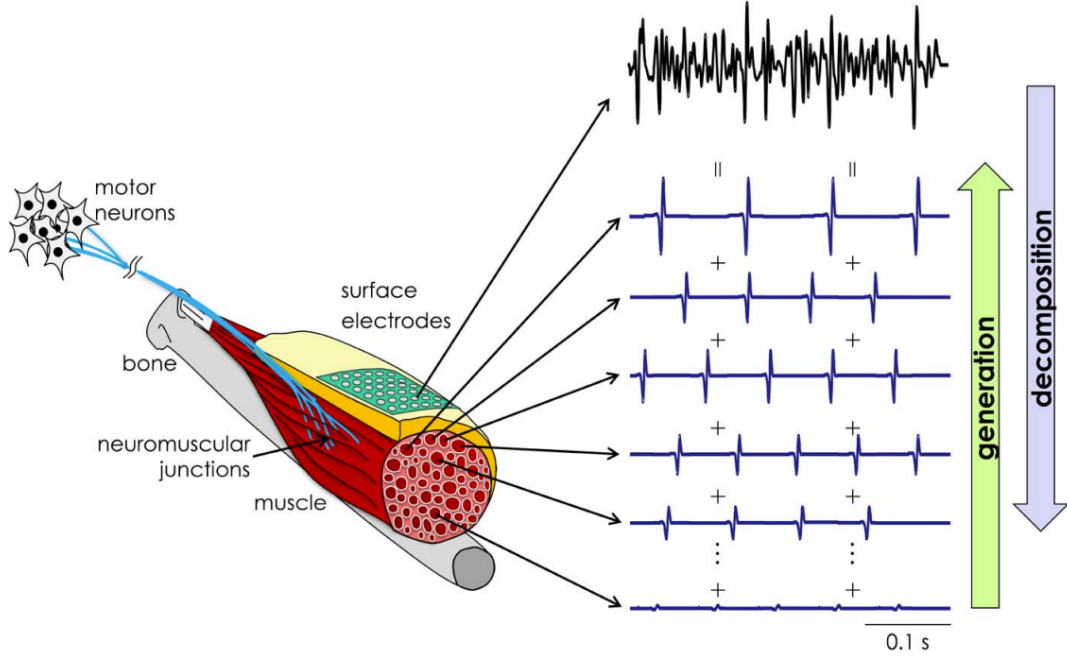


Figure 2.3: A visualization of the linear convolutive mixture data model typically assumed in motor unit decomposition (figure adapted from [48]).

conditions, the **MUAP** of each **MU** is constant for a given channel. Given M channels and N **MUs**, EMG at a discrete time point k can be represented for each channel as:

$$y_m[k] = \sum_{n=1}^N \sum_{l=0}^{L-1} h_{mn}[l] t_n[k-l] + w_m[k] \quad (2.7)$$

where $y_m[k]$ is the EMG observation at channel m , $h_{mn}(k)$ represents a **MUAP** of length L , $t_n(k) = \sum_{k_f \in \Psi_n} \delta[k - k_f]$ is the delta spike train representing the firings of **MU** n and Ψ_n is the set of discharge times of **MU** n , and $w_m[k]$ is additive noise. This model assumes that the average number of samples between consecutive spikes is significantly larger than L .

Rather than computing convolutions directly, the same operation can be expressed as a matrix multiplication through the extended matrix representation [70]. First, the extended representations can be defined for the observations ($\tilde{\mathbf{y}}[k] \in \mathbb{R}^{MR \times 1}$), noise ($\tilde{\mathbf{w}}[k] \in \mathbb{R}^{MR \times 1}$), mixing matrix ($\mathbf{H} \in \mathbb{R}^{MR \times N(L+R-1)}$), and spike train vectors ($\tilde{\mathbf{t}}[k] \in \mathbb{R}^{N(L+R-1) \times 1}$):

$$\begin{aligned} \tilde{\mathbf{y}}[k] &= [y_1[k], \dots, y_1[k-R+1], \dots, y_M[k], \dots, y_M[k-R+1]]^T \\ \tilde{\mathbf{w}}[k] &= [w_1[k], \dots, w_1[k-R+1], \dots, w_M[k], \dots, w_M[k-R+1]]^T \\ \tilde{\mathbf{t}}[k] &= [t_1[k], \dots, t_1[k-R-L+2], \dots, t_N[k], \dots, t_N[k-R-L+2]]^T \end{aligned}$$

$$\mathbf{H} = \begin{bmatrix} \tilde{h}_{11} & \cdots & \tilde{h}_{1N} \\ \vdots & \ddots & \vdots \\ \tilde{h}_{M1} & \vdots & \tilde{h}_{MN} \end{bmatrix}$$

and $\tilde{h}_{mn}$ represents a sub-matrix for the **MUAPs** of **MU** n appearing at channel m :

$$\tilde{h}_{ij} = \begin{bmatrix} \tilde{h}_{mn}[0] & \cdots & \tilde{h}_{mn}[L-1] & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{h}_{mn}[0] & \cdots \tilde{h}_{mn}[L-1] \end{bmatrix}$$

Given the extended representation, the extended observations can be represented as a single linear operation:

$$\tilde{\mathbf{y}}[k] = \mathbf{H}\tilde{\mathbf{t}}[k] + \tilde{\mathbf{W}}[k] \quad (2.8)$$

Given T_s samples across over time, the multichannel EMG can be holistically represented as:

$$\mathbf{Y} = \mathbf{H}\mathbf{T}_{ext} + \mathbf{W} \quad (2.9)$$

where $\mathbf{Y} \in \mathbb{R}^{MR \times T_s}$, $\mathbf{H} \in \mathbb{R}^{MR \times N(L+R-1)}$, $\mathbf{T}_{ext} \in \mathbb{R}^{N(L+R-1) \times T_s}$, and $\mathbf{W} \in \mathbb{R}^{MR \times T_s}$. In this representation, R is the extension factor, and represents the number of delayed repetitions included to improve the condition number of the system. The decomposition problem involves the estimation of $\mathbf{T}_{ext}$ given $\mathbf{Y}$ while $\mathbf{H}$ remains unknown. This involves the estimation of the separation matrix $\mathbf{B}_{full}$ that provides an estimate for the sources $\mathbf{T}_{ext}$, such that:

$$\hat{\mathbf{T}}_{ext} = \mathbf{B}_{full}\mathbf{Y} = \mathbf{B}_{full}\mathbf{H}\mathbf{T}_{ext} + \mathbf{B}_{full}\mathbf{W} \quad (2.10)$$

where $\mathbf{B}_{full} \in \mathbb{R}^{N(L+R-1) \times MR}$. Therefore, the upper limit of the performance of the **MU** decomposition is determined first by how well $\mathbf{B}_{full}$ inverts $\mathbf{H}$ ($||\mathbf{B}_{full}\mathbf{H} - \mathbf{I}||$), as well as its effect on the noise component, where the large weights in the separation vectors can result in disproportionate scaling of the noise power. Additionally, as in practice unidentified sources are often treated as part of the noise component, the optimal separation matrix must also be able to suppress activity from unidentified sources.

Ideally, $\mathbf{B}_{full}\mathbf{H}$ approximates a permutation and scaling matrix. In this formulation, $\mathbf{B}$ would recover the sources at all $L+R-1$ possible delays. However, since these delayed versions represent the same underlying source, recovering all of them is redundant. In practice, the aim is to recover each source at a single, optimal latency k_0 . Thus, the reduced separation matrix $\mathbf{B} \in \mathbb{R}^{N \times MR}$ that retains only the rows corresponding to this specific lag is estimated as:

$$\hat{\mathbf{T}} = \mathbf{B}\mathbf{Y} + \mathbf{B}\mathbf{W} \quad (2.11)$$

where $\hat{\mathbf{T}}$ is the estimate of the reduced source representation, such that only a single source vector is estimated per MU.

2.4.3 Optimal Separation Matrix

Using the Wiener-Hopf equation [84], given the sources signals, the linear minimum mean square error (LMMSE) optimal separation vector $\mathbf{b}_n^*$ is estimated for MU n as:

$$\mathbf{b}_n^* = \mathbf{c}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n}^T \mathbf{C}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1} \quad (2.12)$$

where $\mathbf{c}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n}^T$ is covariance between the extended observations and the ground-truth source signal for MU n . In the context of MU decomposition, since the source signal is assumed to be a series of unit spike trains, the covariance term $\mathbf{c}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n}^T$ is effectively the spike-triggered average (STA) estimator [74] scaled by the spike rate of MU n .

In practice, the source magnitude is often ignored, where separation vectors are normalized and estimated sources are post-processed into a series of binary discharge times. Thus, spike-rate scaling is often ignored, and separation vectors are estimated as:

$$\mathbf{b}_n^* = \mathbf{s}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n}^T \mathbf{C}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1} \quad (2.13)$$

where, as in [70], the STA estimator of a window of length R across channels and set of discharge times Ψ_n is defined as:

$$\mathbf{s}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n} = \lim_{\text{card}(\Psi_n) \rightarrow +\infty} \frac{1}{\text{card}(\Psi_n)} \sum_{k_f \in \Psi_n} \tilde{\mathbf{y}}[k_f] \quad (2.14)$$

Since ground-truth MUAPs are known, $\mathbf{c}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_n}^*$ is computed with the central R samples of the MUAP for all channels for each given MU. Therefore, by concatenating all separation vectors, the LMMSE optimal separation matrix for all MUs is estimate as:

$$\mathbf{B}^* = \left(\begin{bmatrix} \mathbf{s}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_1}^T \\ \vdots \\ \mathbf{s}_{\tilde{\mathbf{y}}, \tilde{\mathbf{t}}_N}^T \end{bmatrix} \mathbf{C}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1} \right) \quad (2.15)$$

$$\mathbf{B}^* = \mathbf{S}_{\tilde{\mathbf{y}}}^T \mathbf{C}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1}$$

where $\mathbf{S}_{\tilde{\mathbf{y}}}^T \in \mathbb{R}^{N \times MR}$ is the STA matrix for all MUs.

2.4.4 Blind Source Separation

The majority of MU decomposition methods rely on the above data model, and intrinsically depend on the difference between MUAP spatiotemporal distributions for accurate separation of each MU. Given the spatial lowpass filtering effects of volume conduction, sEMG measurements are primarily explained by superficial MUs with similar MUAP distributions. While this limits the separability of MU activity for a small number of sparsely placed sensors, a simulation study showed that by increasing the density of electrode placement within the same surface area, the spatial distribution of MUAPs exhibited a significantly larger difference across MUs [53]. Thus, leveraging the differences in MUAP distributions at the skin surface, different approaches can be categorized as either template matching or blind source separation (BSS) approaches.

Template matching approaches extend work from iEMG recordings to multichannel sEMG recordings, where the aim is, from distinguishable waveforms, to extract MUAP templates, followed by clustering of MUAP templates to assign detected templates to these waveforms. Typically, these approaches look for time-points with high energy sEMG signals, and use clustering algorithms to find potential candidates of isolated MUAP firings [30]. While these algorithms can be further improved with heuristics such as regularity of firing rates, their dependence on identifying isolated MUAPs significantly limit their use in sEMG applications, where MUAP overlap is high between different MUs, and increases at higher contraction levels.

One more suitable alternative for handling the superposition of MUAPs is the BSS framework. This class of algorithms are used to solve the cocktail party problem where, under the assumption that observations are a linear mixture of latent, unknown sources, the aim is to extract these underlying sources. With only the observations available, the process is "blind" to the structure of the underlying sources and the mixing process. Despite the complexity of BSS, it has been successfully applied to HD-sEMG recordings, leveraging the property that MU discharges are sparsely distributed [70, 19, 116]. Since sEMG is represented as a convolutive mixture (MUAPs) of sparse sources (spike trains), the process of applying BSS to the extended observations is referred to as convolutive BSS.

Convolutive Kernel Compensation

The first successful convolutive BSS algorithm developed for MU decomposition is known as convolution kernel compensation (CKC) [70]. Given the extended observations $\tilde{\mathbf{y}}[k]$, their global activity index $\gamma[k] \in \mathbb{R}$ is computed using the Mahalanobis distance, defined as:

$$\gamma[k] = \tilde{\mathbf{y}}[k]^T \mathbf{C}_{\tilde{\mathbf{y}}}^{-1} \tilde{\mathbf{y}}[k] \quad (2.16)$$

where $\mathbf{C}_{\tilde{\mathbf{y}}}^{-1}$ is the inverse covariance matrix of the extended observations. Under the noise-free assumption, the convolutive mixture representation can be used to define $\gamma[k]$ in terms of MU properties:

$$\gamma[k] = \tilde{\mathbf{t}}[k]^T \mathbf{H}^T (\mathbf{H}^{-T} \mathbf{C}_{\tilde{\mathbf{u}}}^{-1} \mathbf{H}^{-1}) \mathbf{H} \tilde{\mathbf{t}}[k]$$

$$\gamma[k] = \tilde{\mathbf{t}}[k]^T \mathbf{C}_{\tilde{\mathbf{t}}}^{-1} \tilde{\mathbf{t}}[k] \quad (2.17)$$

Thus, the global activity index can be seen as an indicator of global pulse train activity. In **CKC**, instead of directly using $\gamma[k]$, the Mahalanobis distance is computed between two different discrete time points, obtaining the following relationship:

$$\mathbf{v}_{k_0}[k] = \tilde{\mathbf{t}}[k_0]^T \mathbf{C}_{\tilde{\mathbf{t}}}^{-1} \tilde{\mathbf{t}}[k] \quad (2.18)$$

where $\mathbf{v}_{k_0}[k]$ is the activity index between a discrete time-point k_0 and any other time-point k .

This activity index can be interpreted as the degree of similarity between extended observations at two different time-points that is significant relative to the general covariance structure of the observations. Since spike trains are sparse, their extended covariance matrix is assumed to be approximately diagonal (uncorrelated). This new activity index is approximately given by:

$$\mathbf{v}_{k_0}[k] \approx \sum_{j \in G_{k_0}} c_{jj}^{inv} \tilde{t}_j[k] \quad (2.19)$$

where G_{k_0} is the set of source vector indices that are active (non-zero) at time k_0 .

Thus, the primary contributions to this activity index are the scaled spike trains of **MUs** that were active at k_0 . If a single **MU** is active at k_0 , $\mathbf{v}_{k_0}[k]$ is a direct estimator of its source vector. However, this is not practical in real-world contexts, especially at higher contraction levels with many concurrently active **MUs**, breaking the assumption of uncorrelated sources.

To overcome this, the sequence $\mathbf{v}_{k_0}[k]$ is used to identify candidates by determining the time index with large sequence values, as these points must have contributions from at least one of the active **MUs** at k_0 , followed by the selection of a single candidate, termed k_1 . From Equation (2.19), it can be seen by taking the element-wise product between the sequences $\mathbf{v}_{k_0}[k]$ and $\mathbf{v}_{k_1}[k]$, the primary contributions are from **MUs** that are active at both k_0 and k_1 . This new sequence can be used, termed $\mathbf{v}_{k_0, k_1}$, to identify a new time index candidate, referred to as k_2 , and compute the element-wise product $\mathbf{v}_{k_0}[k] \cdot \mathbf{v}_{k_1}[k] \cdot \mathbf{v}_{k_2}[k]$. This process is carried out for a total of N_C time indices, obtaining a sequence $\mathbf{v}_{k_0, k_1, \dots, k_{N_C-1}}[k]$, primarily composed of contributions from **MUs** that are active at all time indices $k_0, k_1, \dots, k_{N_C-1}$. Since sources are assumed to be weakly correlated, by choosing a large enough N_C , contributions can be assumed to come from a single active **MU** with high probability. This is effectively an estimate of the discharge times of a single **MU**, which are used to estimate the **LMMSE** optimal separation vector from Equation (2.15) to further refine the estimates of the underlying discharge times. Thus, under the assumption of sparsity, even when allowing overlap between spike trains, **CKC** can extract the discharge times of single **MUs** without any knowledge of the mixing matrix.

When applied to phenomenological and biophysical **HD-sEMG** simulations, **CKC** was able to accurately decompose a large number of active **MUs**, even when introducing weak correlations be-

tween sources [70]. In a subsequent study, **CKC** was applied to **HD-sEMG** recordings from healthy subjects during isometric contractions at 10% of the **maximum voluntary contraction (MVC)**. Simultaneous **iEMG** recordings were acquired for validating the accuracy of decomposition [72]. **CKC** was able to identify between 6-19 **MUs** for each muscle with a total agreement with **iEMG** decomposed activity of $98 \pm 1\%$, validating its ability to decompose **MU** activity from experimental **HD-sEMG** recordings.

Independent Components Analysis

A technique that is more widely established for general **BSS** applications is **independent component analysis (ICA)**, an **unsupervised learning** method that extracts the underlying sources that are the most independent. The central limit theorem states that the sum of independent random variables of any distribution will tend to a Gaussian distribution. ICA leverages this theorem under the working principle that independent, unknown sources should follow less of a Gaussian distribution than any linear mixture of them. Hence, assuming underlying sources are independent and non-Gaussian, ICA aims to invert the mixing process by maximizing the non-Gaussianity of extracted sources [76].

To maximize non-Gaussianity, functions of the data are implemented that measure the deviation from some property of the Gaussian distribution, referred to as contrast functions. One intuitive example is kurtosis, a scaled fourth-order moment of a probability distribution that measures "tailedness". Since Gaussian distributions are symmetric and have zero kurtosis, this statistical property can be directly used as a learning objective for maximizing source non-Gaussianity. However, kurtosis is sensitive to outliers, and can result in unstable optimization. Thus, more robust functions have been introduced, such as negentropy, which computes the difference in the entropy of a Gaussian distribution with the extracted sources. Since for a fixed covariance structure of continuous random variables the distribution that maximizes the differential entropy is the Gaussian, negentropy increases with non-Gaussianity, only being zero when the extracted sources are Gaussian. Since differential entropy requires a complex estimation procedure, it is often estimated with specific approximation procedures [76].

Rather than jointly optimizing all sources, one of the most common implementations of the framework is the **fast independent component analysis (fastICA)** algorithm, which greedily optimizes separation vectors to extract each source sequentially (as in **CKC**) with an iterative gradient-based optimization method [76]. Typically, a whitening algorithm is applied to the observations prior to optimization to obtain normalized, uncorrelated representations of the observations. **FastICA** uses an efficient estimation of a Newtonian update at every iteration referred to as the fixed-point iteration algorithm until convergence. Once a source estimate is obtained, the separation vectors are re-initialized and algorithm is re-run. To prevent convergence to the same sources as already identified, the extracted sources must be decorrelated with the observations at every iteration. This is typically done with or Gram-Schmidt process [76] or a peel-off approach [116].

The framework can be readily applied to the **sEMG** signals due to the same property of **MU** activity that is leveraged by **CKC**: sparsity. Under the linear convolutive mixture model where sources are represented by spike trains, the probability of exact overlap between sources is negligible, a

manifestation of statistical independence. Additionally, MU activity is also highly non-Gaussian, following a highly leptokurtic distribution (fat tail) due to its spiking nature. Thus, *fastICA* can be applied to HD-sEMG signals, extracting a large portion of MUs with high rates of agreement with simulated signals, and extracting 14 MUs from experimental recordings [19]. However, without simultaneous iEMG recordings, there was no concrete way to evaluate decomposed MUs from experimental data.

Negro et al., (2016) implemented a framework with *fastICA* at its core [116], but with two key additional steps: the use of the K-means++ to cluster real-valued source values into spikes and non-spikes, and a refinement procedure based on the estimate of the LMMSE optimal separation vector from predicted discharges, as done in the original CKC study [70]. This framework was also evaluated on simultaneously acquired iEMG and HD-sEMG recordings, and found not only a large number of MUs with high rates of agreement between the two acquisition modalities (an average of 14 MUs across tasks and subjects), but were able to decompose MUs even at higher contraction levels (50% and 90% MVC), a problem that cannot be solved with simple template matching approaches. Thus, convolutive BSS is a powerful tool to extract the activity of a large number of MUs non-invasively.

2.4.5 From Motor Unit Activity To Motor Intent

α -MNs provide the neural commands for movement generation, hence their activity can be leveraged for estimating motor intent. In particular, one popular use of activity sampled from the same motor neuron pool is to estimate the common synaptic input. The output of a neuron can be treated as a transformation of continuous synaptic current into discrete events (spikes) via a non-linear thresholding mechanism. Since force generation in muscles requires the coordinated firing of individual MUs, the input to motor neurons can be separated into a common input and non-common inputs. Intuitively, as the common input increases, a larger number of MUs will fire within the same time-period, both due to MU recruitment and increasing firing rates. Thus, the magnitude of the common input is correlated with instantaneous firing rate averaged over MUs. On the other hand, the outputs due to independent components are cancelled out due to destructive interference. Thus, this average instantaneous firing rate is proportional to the common input intensity.

To investigate this, simultaneous HD-sEMG and iEMG recordings were acquired from healthy subjects and based on the decomposed MU subset, *principal component analysis (PCA)* was applied to lowpassed spike trains, providing a measure of instantaneous firing rate [115]. Given the low-frequency properties of the muscle fibre twitch, it was expected that the low-frequency components of the neural drive (MU outputs) would be the primary determinant of force generation. While the smoothed spike trains of single MUs were not strong predictors of force, the first principal component of smoothed discharge rates across MUs explained 72% of measured force variance. Additionally, the first principal component explained 44.2% of the variance in the smoothed discharge rates, which was significantly higher than the explained variance of the consecutive components. This finding supported the linear transmission of the common input to the neural drive. Altogether, this implies that the low-frequency components in the common input are

responsible for force generation, later termed the effective neural drive [51].

Assuming a linear transmission of the common input across motor neurons, rather than employing PCA, aggregate spike trains can be directly estimated with an unweighted sum known as the cumulative spike train (CST), providing an estimate of the instantaneous firing rate averaged across MUs. In this context, the significant linear coherence values observed between cortical (EEG) measurements and sEMG envelopes (corticomuscular coherence) were investigated using the CST. Using a collection of perfect integrate-and-fire (PIF) neurons, authors demonstrated analytically that the CST shows a linear relationship with the common input, provided sufficient discharge rates and number of MUs [114]. From simultaneously acquired EEG and HD-sEMG recordings, the CST from 4-5 MUs was shown to be sufficient to reconstruct the observed corticomuscular coherence, also reflecting how common input properties can be estimated from a small subset of extracted MUs. When the coherence of the CST from two subsets of motor neurons from the same motor neuron pool is computed, the coherence values increased significantly as a function of the number of motor neurons [51]. Thus, not only do the lower-frequency components of common synaptic input reflect the effective neural drive, but this common synaptic input, with origins in the CNS, can also be estimated from a small subset of neurons from the motor neuron pool.

2.4.6 Applications in Physiology and Neural Engineering

The ability to extract MU spike trains from HD-sEMG has fundamentally advanced our understanding of the neural control of movement in both health and disease. In healthy ageing, decomposition has highlighted the mechanisms of sarcopenia, distinguishing between the loss of functional MUs and the remodelling of NMJs [129]. In pathology, the analysis of MU populations provides a window into the CNS that global sEMG amplitude cannot. For example, in movement disorders such as Essential Tremor or Parkinson's disease, the coherence analysis of MU spike trains allows researchers to isolate the CNS oscillatory sources of tremor from peripheral mechanical resonances, offering specific biomarkers for differential diagnosis [58].

In the domain of myoelectric control, MU spike trains offer a superior alternative to traditional intensity-based sEMG control. By tracking individual motor neurons, researchers can bypass the amplitude cancellation and cross-talk inherent in global sEMG. Control signals derived from the smoothed discharge rates of populations of MUs were shown to significantly outperform traditional sEMG amplitude metrics in a *gesture*¹ classification task [54]. Discharge rates were also mapped to continuous degrees of freedom, exhibiting a high degree of controllability across subjects while providing means of proportional control. Researchers have also investigated the use of MU firing rates as degrees of freedom, enabling motor complete SCI subjects to control a virtual hand [119].

¹Throughout this thesis, the standard term *gesture* is used as a macro-level class label for configurations of sustained isometric muscle effort rather than physical movement. See the front-matter Glossary for the complete methodological definition.

2.5 Motor Intent Decoding: Machine Learning

2.5.1 Machine Learning

One of the most popular means of estimating motor intent from EMG is through [machine learning \(ML\)](#), which is used to solve tasks (T) that are too complex for fixed computer programs. Over the last two decades, [ML](#) has resulted in widespread applications that are now a part of everyday life. This involves solving scientific problems that have been deemed unsolvable for the last century, as in the case of protein folding prediction [81], to the use of large-language models (LLMs) in business and everyday life [132], among many others. Given the complex nature of predicting motor intent from [sEMG](#), [ML](#) provides numerous frameworks based on concurrent recordings of [sEMG](#) and motor intent. [ML](#) algorithms involve the processing of an example, a collection of features often represented as a vector $\mathbf{x} \in \mathbb{R}^n$, to solve the task of interest.

The three categories of [ML](#) are:

1. **Supervised Learning:** the [ML](#) algorithm experiences a dataset with examples, where each example is also associated with a label, $\mathbf{y} \in \mathbb{R}^m$, also referred to as a target (supervision), where m is the number of quantities to predict. The aim of supervised learning is to optimize a set of model [parameters](#), denoted as θ , to approximate a function f_θ that best maps the input to the desired output according to this distribution. From a probabilistic perspective, given random variables X and Y associated with example and label vectors respectively, this is equivalent to estimating the conditional probability distribution $P(Y|X)$ (also referred to as target concept). In the ideal case, $y = f_\theta(\mathbf{x})$ is obtained [61].
2. **Unsupervised Learning:** the [ML](#) system experiences a dataset with examples and no associated labels, hence no supervision. Probabilistically, the observations X are assumed to be generated by latent variables Z via a generative distribution $P(X|Z)$. The goal of unsupervised learning is often to optimize parameters θ to estimate the posterior distribution $P(Z|X)$, effectively inverting the generative process to recover the latent structure.
3. **Reinforcement Learning:** a framework enables an "agent" to perform the optimal "actions" within an "environment". The optimality of an action is dependent on the goals of the agent, as well the "state" the agent is in within the environment [147].

From a multitude of different categories and taxonomies within the generic [ML](#) tasks, the most relevant to motor intent estimation from [sEMG](#) are:

1. **Classification (Supervised):** In this task, the [ML](#) algorithm has to predict the discrete category that an example belongs to out of k total categories based on example's features. Hence, the learning process optimizes parameters θ of a function $f_\theta : \mathbb{R}^n \rightarrow \{1, \dots, k\}$ that minimizes classification error [61]. This framework has been widely used in computer vision, such as classifying an image [89], or whether two face images belong to the same person facial recognition [148]. It has also been used in prominent applications in [natural language processing \(NLP\)](#) for tasks such as sentiment analysis ("thumbs up" or "thumbs down" movie reviews) [123] and sentential paraphrasing (whether two sentences have the

same meaning or not) [36]. By modelling motor intent as a discrete, finite set of actions, such as wrist/finger gestures, classification algorithms can be trained based on an example of sEMG features [156].

2. **Regression (Supervised):** In this task, the ML algorithm has to predict a continuous quantity based on an example's features. Hence, the learning process optimizes parameters θ of the function $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that minimizes the distance between predictions and labels, where m is the number of real-valued quantities estimated. This framework has also seen a wide array of applications, such as predicting the coordinates of bounding-boxes for object detection in images [133]. Given this framework, the aim of a regression problem in the context of sEMG is to, based on the features of an sEMG example, predict a continuous motor quantity, such as force, torque or joint angles [156].
3. **Representation Learning (Unsupervised/Supervised):** In a representation learning task, f_θ is optimized to map complex inputs to a new representation, referred to as an embedding or latent representation. In a probabilistic sense, this involves the optimization of parameters θ such that $\hat{P}_\theta(Z|X) \approx P(Z|X)$, effectively inverting the generative process. θ is usually optimized to contain certain properties that make them valuable for other downstream tasks. Thus, the outcome of the learning process is a function $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^m$, where the input is an example $\mathbf{x} \in \mathbb{R}^n$, and the output is the latent representation $\mathbf{z} \in \mathbb{R}^m$.

This can be useful for visualising examples with lower-dimensional representations, such as with Laplacian eigenmaps [10], or for obtaining a more useful and holistic representation of a given example, such as the case of Word2Vec, an algorithm that maps a word to a numerical vector representation that encodes the relationship of a given word to all others in a corpus [109]. Such functions are often used to obtain meaningful representations for downstream tasks. In the context of sEMG, representation learning is not only used for dimensionality reduction, but is also used to extract physically meaningful quantities within the signals, such as MU activity [19].

In this section, I focus on the predominantly used classes of algorithms in motor intent estimation, namely supervised and unsupervised learning.

Performance & Generalization

To evaluate the capabilities of an ML algorithm, a quantitative performance measure (P) must be defined as closely related to the downstream tasks as possible. The aim of optimizing a model is to maximise its performance on all possible examples, while only having access to a finite dataset. In supervised learning, from a probabilistic perspective, a dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{|D|}$ is considered to be a collection of independently and identically distributed (iid) samples from an underlying data generating distribution $P(X, Y)$, such that $D \sim P(X, Y)$. However, if the dataset size $|D|$ is limited or the iid assumption is violated, the available dataset may not be representative of $P(X, Y)$. In such cases, the learned conditional distribution $\hat{P}_\theta(Y|X)$ is not a good estimate of the true $P(Y|X)$. In other words, models that are optimal given D may not exhibit high performance across all samples from $P(X, Y)$. Thus, the aim of ML is to train a model that performs optimally on examples

unseen during training, a property known as *generalization*. The same framework can be extended to unsupervised learning by considering instead the estimate of $P(Z|X)$.

To track the ability of a model to generalize, the dataset is often split into two principal sub-datasets: the training set (D_{train}) for optimizing parameters θ , and the test set (D_{test}) for evaluating generalization performance. From these two sub-datasets, two generalization failure modes of ML can be identified. Underfitting occurs when the ML algorithm exhibits low performance on both training and test sets, while overfitting occurs when the performance measure is significantly lower for the test set than for the training set. The capacity of a model is associated with its hypothesis space: the extent of functions the model can fit to the training dataset. With low capacity, the model struggles to fit the dataset, while large model capacity increases the probability of learning functions that exhibit good training performance, yet low test performance.

One paradigm for improving generalization is known as regularization [61], which includes all techniques that use *inductive biases* (i.e. assumptions about the function to be modelled) to constrain the hypothesis space of the ML model. One classic example known as weight decay or L2-regularisation, which constraints the optimization problem to minimize the magnitude of all individual model parameters [61]. This constrains the hypothesis space to functions with no single model parameter that is too large, leveraging the assumption that generalizable models should not be overly dependent on a single or few features. Another popular regularization method is known as data augmentation, which artificially augments the training dataset by applying label-invariant transformations to examples. An example of this is to randomly translate/rotate images during the training of image classifiers, depending on the inductive bias that the category of an object in an image should be invariant to its position and orientation in the image [89]. In this context, the hypothesis space is constrained to functions that are invariant to translations and rotations of the objects in the image.

During training, the model parameters θ are optimized given D_{train} , with the goal of achieving the optimal performance on D_{test} . However, ML models are often associated with parameters that are not updated during training, referred to as hyperparameters. For instance, the degree of L2-regularization, the number of trees in a random forest (see Section 2.5.3) or the number layers in a neural network (see Section 2.5.6) are chosen before training. Typically, while the optimal hyperparameters can be constrained to a small subset of candidates, these parameters must also be optimized. These parameters cannot be optimized with the training data, and if the test set is used, the test examples cease to be "unseen" examples, increasing the chances of overfitting. Thus, it is common practice to split the training set into two further subsets: a training set and a validation set. With this dataset split, the training set is used to optimize the model parameters, the validation set is used to optimize the hyperparameters, and the examples in the test set remain unseen for measuring generalization performance.

2.5.2 Surface EMG-Based Machine Learning Tasks & Applications

Classification

For classification, sEMG signals are acquired from subjects while performing multiple **repetitions** of a discrete set of movements given visual instructions, usually interleaved with rest periods (no movement). Every sEMG sample is then associated with a label based on the visual instructions displayed at the same time instance. From this dataset, classifiers can be trained by extracting feature vectors $\mathbf{x} \in \mathbb{R}^n$ from every sEMG sample or from a contiguous time-window, and optimizing θ to obtain a mapping from these examples to a target/labelled discrete movement, treated in this context as classes.

For the upper-limb, a traditional framework is hand gesture classification, which involves mapping multichannel sEMG signals to the sustained motor commands of different finger, hand and wrist movements [5, 1, 38, 79], referred to **gestures** or **poses**. In lower-limb tasks, classification can also be used to map sEMG signals to motor commands for a discrete number of locomotion modes during walking [75]. While it is relatively easy to acquire labels for classification tasks from visual cues or instructions, sEMG classification performance is often significantly impacted by movement technique, and does not naturally provide measures of the intensity of the movement being performed.

Applications: The primary downstream applications of supervised sEMG classification involve myoelectric control, where the predicted classes or joint angles/forces (motor intent) enable the user to control external devices. One canonical example is the myoelectric control of prosthetic limbs. For instance for upper-limb amputees, multichannel sEMG recordings can be acquired from the residual forearm muscles to predict movement intent, with sEMG acquired during attempted movements given visual cues [6]. With lower-limb amputees, classification methods have also been developed to predict motor intent from the residual leg muscles for lower limb prosthetic control [75]. Myoelectric control has also been used to control robotic exoskeletons, which can provide power assistance during movement generation for activities of daily living and rehabilitation, mapping the motor intent of the user to force commands [110]. sEMG classification has also been extensively investigated for the myoelectric control of **virtual and augmented reality (VR/AR)** systems, allowing subjects to interact with virtual systems directly from motor intent, eliminating the need for mechanical interfaces [138, 143, 83].

Beyond myoelectric control, supervised learning models can also be used for neurorehabilitation and clinical assessment. The Fugl-Meyer Assessment (FMA) is a standard framework that involves numerous time-consuming tests carried out by clinicians to assess upper-limb stroke impairments. Supervised ML models have been trained to map sEMG recordings (along with other motor information) during a small set of movements to Fugl-Meyer scores, significantly reducing clinician time [120, 2]. Another ML model was trained to map sEMG recordings acquired from various trunk muscles to different forms of trunk compensation for stroke patients, with the aim of helping stroke patients correct their posture during rehabilitation [98].

Regression

For regression, sEMG signals are acquired from subjects performing different movements, while simultaneously acquiring signals from sensors that provide direct measurements or proxies of two primary physical quantities: joint angles and force/torque. By dynamically measuring these quantities for the task-relevant body parts, continuous measurements of the movements performed by subjects are obtained. Since these quantities are both directly associated with the contraction levels of relevant muscles, by simultaneously acquiring these continuous quantities and multichannel sEMG recordings, regression models can be trained to predict them from sEMG examples. Typically, marker-based [82] or markerless [138] motion capture systems can be used to acquire joint angles from the upper and lower limbs. Upper-limb hand kinematics can also be acquired through *Cyberglove* systems, which are gloves that use resistive bend-sensing technology [5]. Force/torque measurements can be acquired with strain gauge load cells [5].

Applications: Given appropriate sensor recordings, regression-based models can be used as an alternative framework for the aforementioned tasks for classification. Although regression-based prosthetic control is more nuanced, since direct force and joint angles cannot be acquired for the missing limb, mirrored bilateral training has been used, where sensors are placed on the non-affected limb and subjects are asked to perform movements using both limbs, and measurements from the non-affected limb are used as the labels [117].

Unsupervised Learning

One standard use of unsupervised learning in sEMG is for dimensionality reduction, where the number of features in the dataset is decreased for a downstream supervised learning task [40]. However, by leveraging inductive biases of the underlying motor control and biophysics generation of sEMG, examples can be mapped to new representations that correspond to some anatomical or function property of motor control. Under the assumption that motor control uses a low-dimensional basis to generate a diverse set of movements, unsupervised learning methods have been used to estimate these low-dimensional components, referred to as *synergies* [12, 149]. Another widespread use of unsupervised learning is found in MU decomposition, where sEMG examples are processed into the activation of distinct underlying MUs [19, 116].

Applications: As in supervised settings, unsupervised learning can also be used directly or indirectly for myoelectric control. When extracted synergies were provided as control signals, subjects could intuitively activate them to control a virtual cursor on a screen [12]. Similarly, extracted MU activity from decomposition has been shown to enable intuitive control, whereby subjects can learn to modulate the firing rate of individual MUs, or more intuitively, using the aggregate firing activity of all MUs from the same muscle as a proxy of muscle activation (see Section 2.4 for more information on MU decomposition).

Another important benefit of unsupervised learning is improved understanding of the neural control of movement. Given the large number of degrees of freedom involved in human movement, it has been hypothesised that the nervous system uses low-dimensional representation of joint muscle activations for efficient motor control. By applying different matrix factorization methods

to **sEMG** recordings from healthy subjects, a low-dimensional number of components were able to effectively capture the structure of the data [12]. In stroke-affected subjects with mild motor impairments, it was shown that while extracted synergies were comparable between the affected and non-affected upper limbs, the modulation of these synergies varied significantly. This reflects how, under the view of muscle synergies, the **CNS** control of these underlying synergies is affected, providing a potential tool for stroke clinical assessment [22].

2.5.3 Traditional Machine Learning

Feature Engineering

Traditionally, an **ML** pipeline would be composed of two distinct components: manual feature extraction, and the training of an **ML** algorithm that uses those feature vectors as examples to solve a particular task. The process of constructing a feature set manually is known as feature engineering where, especially relevant in supervised learning, human experts build functions of the data to return features that are known to be predictive of the target. This is an effective application of domain-expert inductive bias to simplify the original (raw) data structure and reduce its feature dimensionality. For instance, in **NLP**, Bag-of-Words (BoW) has been a simple, yet powerful feature extraction framework, where the count of present words or stems of words are used as independent features in an example vector [80]. Given a dataset of documents, models are trained to learn relationships between feature vectors based on the frequency of key words for the desired task or objective, such as topic classification. In the field of computer vision, image features are designed by either aggregating pixel-level quantities over the whole image, such as computing the average of channels in different colour spaces, or by manually designing digital filters or kernels that extract meaningful spatial information, such as gradient estimators. A canonical example of feature engineering in computer vision is the scale-invariant feature transform (SIFT), which identifies key points in an image and extracts rotation and scale-invariant properties from these points based on local gradients [96].

Linear Models

After feature engineering, there are numerous classes of supervised **ML** algorithms that can process examples, now in a numerical vector form, to the desired target. One class of algorithms is known as linear models, which assume a linear relationship between the features and the target. All such models assume the form $y = \mathbf{w}^T \mathbf{x} + b$, where $\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$ are the model parameters (θ) that completely determine the mapping of the model. These models are trained by optimizing their parameters to fit a hyperplane to the feature vector space. In regression, the optimization objective is often to fit the hyperplane through the points that returns the lowest mean-square error across all examples in the dataset. In classification, the aim instead is to fit the hyperplane between the points to give the best class separability. For these canonical learning objectives, linear models have the mathematical guarantee of convexity: the objective as a function of the model parameters has a single, unique global optimum [32].

For unsupervised learning, linear methods are effectively different means of matrix factorization, where the aim is to learn a linear projection of the original features in a (typically) lower-

dimensional subspace, under the assumption that the examples can be modelled as:

$$\mathbf{x} = \mathbf{H}^T \mathbf{z} + \mathbf{w} \quad (2.20)$$

where $\mathbf{z} \in \mathbb{R}^m$ is the lower-dimensional representation of example $\mathbf{x} \in \mathbb{R}^n$, often referred to as the latent representation, $\mathbf{w} \in \mathbb{R}^n$ is additive noise, and $\mathbf{H} \in \mathbb{R}^{m \times n}$ is the generative matrix that linearly combines the lower-dimensional components to return the original examples.

While using the same data model, different methods learn different generative matrices depending on the learning objective for the downstream task. In [PCA](#) [32], the aim is to find the m -dimensional latent representation with orthogonal components that minimize the reconstruction error (remains as similar to the original examples as possible). [PCA](#) is useful for reducing the dimensions of the feature set for downstream supervised learning tasks [40], as well as for visualizing high-dimensional feature vectors in a low-dimensional subspace. Another method is the previously discussed (see Section 2.4), which learns the generative matrix by maximizing the independence of components in the latent representation. Another matrix factorization technique that is commonly applied to [sEMG](#) signals is [non-negative matrix factorization \(NMF\)](#), which also aims to minimize reconstruction error, but instead of orthogonal components, the generative matrix and input examples are constrained to be positive. Being purely additive, [ML](#) can extract more interpretable latent representations, such as representing images of human faces as the sum of different facial components [91].

While linear methods are still widely used and the assumption of linearity can work more generally for monotonic relationships, it still imposes significant constraints on the learnt relationships, constraining feature engineering methods to provide features that exhibit linear relationships with the target.

Non-Linear Models

To overcome the constraints of linearity, a simple solution is to expand the feature vector by adding non-linear transformations of each feature. For example, a simple feature vector $\mathbf{x} = [x_1, x_2]^T$ can be extended to obtain $\tilde{\mathbf{x}} = [x_1, x_1^2, x_2, x_1 x_2, x_2^2]^T$. From this feature representation, linear models can be trained, where each new transformed term is just treated as an independent feature of the model. However, this explicit expansion results in the curse of dimensionality: the dimension of the extended vector grows rapidly relative to the input size n . For high-dimensional data, this imposes prohibitive computational costs and increases model capacity, making it prone to overfitting.

One solution, known as the kernel trick, can be used in different linear geometric models while maintaining the same number of parameters and computational complexity. This method, especially popular within [support vector machines \(SVMs\)](#) and support vector regression (SVR) for classification and regression respectively, can use polynomial kernels, as well as kernels that add other linearities, such as radial basis function (RBF) kernels [32]. Additionally, by scaling the number of parameters with the number of support vectors rather than the feature dimension, [SVMs](#) became dominant in complex domains with high-dimensional feature vectors.

For text categorisation, an approach using a BoW feature representation achieved performances with an SVM and an RBF kernel that significantly outperformed other classifiers [80]. In human detection in computer vision, a histogram-of-gradients (HOG) method was used to extract a high-dimensional feature vector with local gradient information from different image regions [28]. An SVM using an RBF kernel was able to handle high-dimensional examples efficiently, outperforming alternative classifiers. Despite these successes across domains, kernels still require explicit design choices while still retaining the limitations of manual feature engineering for their linear counterparts.

Decision-Tree Based Models

Another popular class of ML algorithms is decision trees. Such models recursively partition the feature vector space into sub-regions based on a hierarchical set of numerical thresholds. While decision tree classifiers assign predictions as the most frequent class within specific sub-regions, decision tree regression assigns a continuous prediction as the aggregate value of training examples within the sub-region [88]. Crucially, the sequential partitioning on individual features based on certain split criteria allow decision trees to naturally ignore features that are not useful, without having to manually select them. The model capacity can be improved by increasing the number of allowed partitions, with every prediction having an interpretable and traceable number of steps for the final decision-making. Additionally, by optimising the partitions, decision trees are able to learn non-linear boundaries. Categorical features can also naturally be processed by comparing the feature value to a fixed set of values. However, due to the hierarchical decision-making involved, decision trees are inherently unstable to small changes in the training data.

One of the most prominent ways of overcoming the limitations of decision trees is to use ensemble models that combine the use of multiple models of the same class. For instance, random forests aggregate the outputs across simple decision trees trained on random training data and feature subsets. By aggregating across many simpler models, the final output is significantly more robust than its constituents and its training and inference can readily be parallelized. This made random forests a strong candidate for pixel-level human pose classification for real-time applications where a classifier was needed that was computationally efficient while still effective at handling non-linearities and high-dimensional feature sets [141]. Another form of ensembles is tree-boosting, which trains decision trees sequentially, with consecutive trees being optimized on the errors of previous trees. XGBoost, an efficient implementation of gradient-boosting, was used in a broad range of research and industry problems, used as the classifier in 17 out of the 29 winning solutions in Kaggle data science challenges from 2015 [20].

Despite the success of feature engineering and the models discussed across disciplines, this paradigm imposes a static representation of the data from which models are trained for a dataset. By summarizing raw inputs into simplified scalars, such as word counts or image gradients, there is an irreversible information bottleneck, preventing the model from learning relationships with features of the data that were not included by the human expert. This approach also suffers from a dependency on domain expertise: feature extractors are difficult to build, brittle, and often not directly transferable across domains. Thus, feature engineering is prone to neglect relevant information in

raw data and ultimately limits the capacity of the model trained to solve complex tasks.

2.5.4 Surface EMG-Based Traditional Machine Learning

After sEMG signals are acquired and appropriately filtered, feature vectors must be constructed to train traditional ML models. For supervised learning, while typically labels are available for every sEMG sample (discrete movement performed in classification or joint angle/force measurement for regression), unprocessed sEMG samples are not strong predictors. Thus, large bodies of research have been dedicated to feature engineering, whereby leveraging inductive biases of the nature of sEMG biophysics generation, the aim is to manually extract a set of task-relevant feature vectors.

Time-Windowing

The most common framework of feature extraction in sEMG involves time-windowing: extracting feature vectors from short, contiguous time-windows of sEMG, where the label associated with each feature vector is either the label in the centre of the time-window or some representative label statistic for the given window (the mode of labels for classification and the mean/median for regression). Time-windows are dependent on two primary hyperparameters: **window size** (w_{size}), determining the number of seconds/samples of each window, and **window stride** (s), which determines the number of seconds/samples between two consecutive windows, where there is a window overlap whenever $w_{size} < s$. While larger window sizes provide more samples for estimating robust features, they also introduce larger delays, with the effective delay introduced being approximately $\frac{w_{size}}{2}$. By processing consecutive windows, ML models produce a continuous prediction stream. Since sEMG ML models are typically used for real-time interfacing, the acceptable delay in the prediction stream (determined by the window size and processing time) is set to be imperceptible to the subject, typically set to be approximately 300 ms [39]. In classification tasks, a majority voting windowing operation can be applied, which returns the most frequent class within a time-window, at the cost of increased delay.

Time-Domain Features

The most common approach to manual feature extraction involves the estimation of time-domain (TD) properties for each window across all channels. Since the aim is to extract multiple complementary features, the feature extraction process generates $\mathbf{x} \in \mathbb{R}^{n_{features} \cdot n_{channels}}$ as the examples, where $n_{features}$ is the number of properties extracted and $n_{channels}$ is the number of sEMG channels acquired. A canonical feature set for sEMG-based ML models, commonly referred to as the TD feature set, involves the estimation of four simple time-domain statistics: mean absolute value (MAV), zero-crossings (ZC), slope sign changes (SSC) and waveform length (WL) [39].

The MAV feature measures the average signal magnitude within the time-window, and has been shown to vary monotonically with the contraction level of the relevant muscle [95]. The three other features capture complementary how rapidly sEMG is fluctuating, acting as proxies for the frequency content in the signal. By extracting TD features from four sEMG channels with anatomically-oriented placements along the forearm, a linear discriminant analysis (LDA) model

was used to map the features to four discrete wrist movements, achieving test accuracies of over 90% on held-out repetitions of the same session. This suggests that domain-relevant time-domain features are sufficient for linearly discriminating between a small set of discrete movements.

Time-Frequency Representations

However, the properties in the **TD** feature set represent average properties within a time-window, best suited for making predictions during sustained contractions where **sEMG** activity is assumed to be stationary. Hence, to capture transient properties during dynamic contractions, time-frequency representations were introduced. Using time-windows from **sEMG** signals measured from two channels (biceps and triceps), test accuracy on a held-out repetition from the same recording session of four discrete movements during movement onset was significantly improved relative to **TD** features when using time-frequency representations and **PCA** dimensionality reduction [40].

Inspired by these findings, another popular feature set was introduced, referred to as the NinaPro feature set, which on top of **TD** features, also includes the marginal discrete wavelet transform (mDWT), the **root mean square (RMS)**, and EMG histogram features [5]. This feature set exhibited classification accuracy improvements across a variety of classifiers and datasets from the publicly available NinaPro database relative to the **TD** features. Furthermore, this study also obtained best performances with random forests and **SVMs**, suggesting that the choice of classifier can significantly impact prediction performance.

Extracting Temporal & Spatial Properties

Numerous feature sets have been constructed to leverage inductive biases based on temporal **sEMG** properties. These features have been typically extracted independently for each channel and concatenated into one augmented feature vector, depending on the downstream **ML** models to learn relationships between channels. While time-based feature extraction has resulted in successful **ML** models, they do not leverage inductive biases of the spatial structure of **sEMG** signals to improve generalization. Thus, different feature engineering methods have also been developed to leverage the spatial structure of multichannel **sEMG** recordings. Typically, these require some degree of placement regularity, as is the case in **HD-sEMG** arrays or when placing electrodes evenly along the circumference of the relevant limb.

The most direct form of leveraging spatial inductive biases from regular arrays of multichannel **sEMG** is by processing **sEMG** samples or time-window features across channels as images. In this context, each **sEMG** electrode corresponds to a different pixel in the image, and different extracted features can be treated as the colour space of the image. Different methods from traditional computer vision have been implemented to determine regions of the grid that exhibit the largest signal power (captured by **RMS** features) during contractions, such as the Watershed algorithm [1]. Image-based processing of **sEMG** grids introduces spatial inductive biases that can improve generalization, provide interpretability by enabling the visualization of the **sEMG** distribution over space, and allow the use of well-established techniques from computer vision.

Beyond image-based techniques, another technique that processes regular-arrays of **HD-sEMG**

is the variogram from geostatistics [144]. instead of using different feature vectors to be concatenated, variograms treat pairs of channels as realisations for estimating the degree of spatial correlation in the grid. While variograms showed slightly worse performance compared to **TD** features for a held-out repetition of the same electrode placement, it showed significant improvements over other feature extraction methods when applied to spatially translated versions of the same **HD-sEMG** array, reflecting its robustness to electrode displacement.

Synergy Extraction Through Matrix Factorization

Matrix factorization methods have been extensively applied to **sEMG** signals to extract low-dimensional representations, referred to as synergies. While different matrix factorization methods have been used, pipelines normally extract the **sEMG** envelope from each channel (lowpass filtered and rectified), apply **ML** to each time-point sequentially, extracting time-varying low-dimensional components that can additively reconstruct the observed **sEMG** signals [12]. **ML** has also been applied to the output of **MU** decomposition to extract information at the **MU** level, rather than a surface potential level. In this context, binary spike trains are constructed from **MU** discharge times and lowpassed, providing a continuous estimate of instantaneous firing rate. After **ML**, the extracted motor neuron synergies accurately reconstructed the instantaneous firing rates with a ten-fold decrease in dimensionality, further supporting the concept of low-dimensional movement control [149]. While similar in dimensionality to the extracted muscle synergies, motor neuron synergies were more predictive of the generated force, supporting the extraction of synergies at the **MU** level. Motor neuron synergies a great example of the combination of two linear unsupervised learning methods (**ICA** and non-negative matrix factorization) to extract structurally relevant information of motor control.

2.5.5 Gradient-Based Optimization

Training **ML** models, whether supervised or unsupervised, can be summarized as finding the best set of model parameters (θ^*) for a given dataset D [32]. This process is done through the optimization of an objective function, a function of the model parameters that quantifies how well f_θ approximates the underlying conditional distribution (e.g., $P(Y|X)$ or $P(Z|X)$). While optimization methods may either maximize or minimize the objective function depending on the task, it is common to only consider minimization, where a maximization objective is often reframed as the minimization of the negative of the objective function. During minimization, the objective function can also be referred to as the loss function (or loss for short), as well as a cost function or error function [61].

One simple framework of optimization methods is known as brute-force search algorithms. This class of algorithms uses different strategies to naively sample a set of possible parameters and evaluate the loss function on a subset or all training examples, returning the sampled set of parameters that attained the lowest loss. Algorithms such as grid-search, random search, and **Latin Hypercube Sampling (LHS)** have been used extensively and are relatively simple to implement [101]. While practical for low-dimensional parameter spaces where sufficient sampling density can be guaranteed, these methods suffer from the curse of dimensionality: the search space vol-

ume grows exponentially with the number of parameters. Thus, [ML](#) more complex models and high-dimensional feature sets require more effective optimization methods.

For linear models, the loss function is often convex and differentiable, allowing for closed-form analytical solutions as for linear regression and guaranteed convergence for [SVMs](#). Other algorithms, such as decision trees employ greedy heuristic search. Greedy optimization methods find the best model update at every step independently, aiming to incrementally approach the global optimum [32]. For decision trees, rather than considering all possible partitions of the data for all feature combinations, for each split the decision tree considers one optimal split along a single feature dimension for the chosen loss function (e.g. purity). While significantly less complex, greedy-optimization of decision trees do not have any mathematical guarantees of convergence to the global optimum. This is another reason that makes decision trees unstable, further supporting the use of ensemble models that leverage multiple decision trees [88].

One popular class of iterative optimization algorithms is known as gradient-based algorithms. When the loss is a smooth, differentiable (or [sub-differentiable](#)) function of the model parameters, gradient information can be computed at every set of model parameters based on the available dataset and use this information to iteratively approach the global optimum. [gradient descent \(GD\)](#) uses the first-order gradients to find the direction of steepest decent, with each step moving the parameters in this direction, incrementally decreasing loss estimates. Since first-order gradients only provide information regarding the direction of steepest descent, they are multiplied with a positive scalar α referred to as the step size or learning rate. The learning rate is often treated as a hyperparameter of the optimization objective. When the first-order gradients are computed at every model update step to obtain the exact gradient information, the [GD](#) update is given by:

$$\theta_k = \theta_{k-1} - \frac{\alpha}{N} \sum_{i=1}^N \nabla_{\theta} \mathcal{L}(\mathbf{x}_i) \quad (2.21)$$

where θ_k are the model parameters at discrete update step k , N is the number of examples in the dataset, and $\nabla_{\theta} \mathcal{L}(\mathbf{x}_i)$ is the gradient of the loss function $\mathcal{L}$ with respect to model parameters θ evaluated at the example $\mathbf{x}_i$.

While [GD](#) is effective for small datasets and number of features, re-computing the gradient at every update can be too computationally intensive for more complex tasks. Thus, [stochastic gradient descent \(SGD\)](#) was introduced [134], where instead of computing the exact gradient at every update step, the gradient is estimated for a randomly sampled example $\mathbf{x}_i$, such that the [SGD](#) update is:

$$\theta_k = \theta_{k-1} - \alpha \nabla_{\theta} \mathcal{L}(\mathbf{x}_i) \quad (2.22)$$

While the original [SGD](#) implementation is significantly more efficient, gradient estimates from a single example may be poor estimates of the true gradient, potentially slowing down or even preventing optimization. Thus, especially popular in modern [ML](#), mini-batch [GD](#) was introduced, which estimates the gradient at every parameter update using a randomly sampled subset of the

dataset, referred to as batch, providing the following update:

$$\theta_k = \theta_{k-1} - \frac{\alpha}{B} \sum_{i=1}^B \nabla_{\theta} \mathcal{L}(\mathbf{x}_i) \quad (2.23)$$

where B is the selected batch size, a hyperparameter of the model, such that $B < N$.

In deep learning models, all based on the optimization of smooth, sub-differentiable functions, use variants of **SGD** as the class of optimization of choice. Second-order gradient-based optimization methods are used extensively in multiple different applications, such as in **fastICA**, which is commonly used for **MU** decomposition [19, 116]. These methods leverage the inverse of the Hessian matrix (second-order partial-derivatives) to provide local curvature information during optimization, enabling the dynamic and adaptive estimation of the optimal learning rates for each model parameter at every update step.

While second-order methods can significantly improve the number of steps to convergence, the inverse of the Hessian matrix scales cubically with the number of parameters, preventing its widespread use in practical deep learning applications, which often depend on millions or billions of parameters. To obtain the computational simplicity of **SGD** but leverage local curvature information to adapt step sizes, numerous second-order gradient approximation algorithms were developed, such as Adam [87]. By approximating the Hessian matrix as a diagonal matrix, Adam cheaply approximates local curvature information and dynamically adjusting the learning rate of individual model parameters, decreasing the number of steps required for convergence and greatly simplifying the process of finding the optimal learning rate.

2.5.6 Deep Learning

Fully-Connected Neural Networks

deep learning (DL) is a subset of **ML** that uses **neural networks (NNs)** with multiple sequential layers to process examples. **NNs** are **ML** algorithms with architectures inspired by biological neurons. In modern **DL**, a neuron is a mathematical function with weights that compute a linear combination of the input feature vector (constituting the input layer) with an added bias term followed by a fixed non-linear transformation of this output, such that the function of the neuron f_{neuron} is given by:

$$f_{neuron}(\mathbf{x}) = \phi(\mathbf{w}^T \mathbf{x} + b) \quad (2.24)$$

where ϕ is the chosen non-linear transformation and its input is of the same form as for linear models (see Section 2.5.3).

The inspiration from the biological neuron comes from its ability to process multiple simultaneous inputs through its dendrites, and generate an **AP** when the aggregation of these inputs exceeds the activation threshold of the neuron [61]. Thus, the non-linear transformation, referred to as an

activation function, is usually formulated to introduce non-linearity, often mimicking the biological 'all-or-nothing' firing rate by saturating at extremes or activating only above a threshold. In NNs, multiple neurons can be used to obtain a new intermediary vector representation of the input features, such that:

$$h(\mathbf{x}) = \phi(\mathbf{W}^T \mathbf{x} + \mathbf{b}) \quad (2.25)$$

where $\mathbf{W} \in \mathbb{R}^{n \times m}$ is the matrix where each column represents the weight vector associated with a specific scalar output, $\mathbf{b} \in \mathbb{R}^m$ is the bias vector added to the vector output, and $h(\mathbf{x}) \in \mathbb{R}^m$ is the new representation, known as a hidden layer. The dimensionality of the hidden layer m is typically treated as a hyperparameter of the NN.

Either another set of weights and biases are applied to obtain the output vector from the final (output) layer, or this hidden layer can be used as the input feature vector to another set of neurons, obtaining consecutive hidden layers. Thus, multilayer NNs can theoretically have any number of intermediary hidden layers before reaching the output function. Within the context of NNs, scalar features in every layer are referred to as nodes. In the above formulation, the NN is referred to as fully-connected, implying that every node from the previous layer is used in the computation of every node in the consecutive layer.

While individual neurons are only able to model a restrictive subset of functions between the input and the desired target, multilayer NNs were theoretically shown to be universal function approximators: the model has enough capacity to represent any mapping between the input features and the target [73], with increasing model complexity by increasing the hidden layer dimensions. Furthermore, each hidden layer can be interpreted as a new latent feature representation of the original input. Since the mappings between consecutive layers are learned during optimization, NNs can be interpreted as a method of automatic feature extraction, with hidden layers progressively obtaining more task-relevant features until reaching the final layer. The final layer returns a real-valued vector (or scalar) that is used in the downstream task. While in regression, the output is directly the quantity of interest, to obtain a prediction from classification, the model returns a real-valued vector $\mathbf{z} \in \mathbb{R}^k$ for the k classes, representing unnormalized scores referred to as logits. From $\mathbf{z}$, the estimate of class probabilities is obtained using the softmax function, such that:

$$p(\mathbf{z})_i = \text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (2.26)$$

where $p(\mathbf{z})_i$ is the probability estimate for class i and the final class is chosen as $\hat{c} = \arg \max_i (p(\mathbf{z})_i)$.

From the output, for unsupervised and supervised learning problems, a loss function is used to quantify the quality of the output. By using sub-differentiable non-linear activation functions, an analytical relationship between the loss function and every weight and bias in the NN is guaranteed. Using the differentiation chain rule, the gradients of the loss function are obtained with respect to final hidden layer activations and the weights and biases that process them to obtain the final output. From these gradients, the derivative chain rule can be used again to obtain the

gradient of the loss function with respect to the activations and parameters of the previous layer. In backpropagation, the canonical optimization method in modern DL, the derivative chain rule is applied sequentially to obtain gradients of all model parameters with respect to the loss function, starting from the final hidden layer all the way to the first [137].

From this information, gradient-based optimization can be applied to all the parameters in the NN. Using SGD, multilayer NNs were able to learn functions shown to be theoretically impossible for individual neurons. However, the non-linear activation functions result in a highly non-convex optimization problem, removing any mathematical guarantees of a single global optimum as with traditional ML approaches, and along with it guarantees of successful optimization. Together with the significant computational complexity of backpropagation in large NNs, DL methods were shadowed by competing ML methods.

Despite the theoretical complexities, multiple advancements in DL were made to address the practical issues observed during training. One significant improvement in gradient-based optimization of NNs was the introduction of momentum, which used an accumulated velocity vector in the direction of steepest descent during gradient updates and helped prevent NNs from becoming trapped in sub-optimal local optima or saddle points [146]. Along with different parameter initialization schemes, great strides were made towards NN optimization. Beyond non-convexity, another issue of backpropagation that required great attention was the vanishing gradient problem: as gradients are backpropagated to previous layers, the gradient of the loss with respect to the parameters progressively decreased, providing near-zero updates for earlier layers. One significant improvement to the issue of vanishing gradients was the introduction of the Rectified Linear Unit (ReLU) as an activation function. Prior to ReLU, functions used to capture the "all-or-nothing" firing of neurons, such as the sigmoid function, compress the activity of the neuron within the range of $h(\mathbf{x}) \in [0, 1]$. ReLU, instead, is a piecewise function such that:

$$ReLU(z) = \begin{cases} 0 & z \leq 0 \\ z & z > 0 \end{cases} \quad (2.27)$$

for some real-valued scalar quantity z .

As with previous activation functions, ReLU is zero below a certain threshold. However, by scaling linearly with the input above this threshold, it allows large contributions to remain large, significantly increasing the possible gradients that could reach the earlier layers, all while being a sub-differentiable function that can still be used during gradient-based optimization [112]. Another advancement was the gradual adoption of graphics processing units (GPUs) for accelerated training and inference of NNs. By enabling the successful training of a deep NN with millions of parameters, GPUs helped fully-connected NNs to reach SOTA performance on the handwritten digit classification benchmark (MNIST) for the first time [24]. One of the foundational successes of DL models was the use of GPUs and momentum gradient-based optimization to train deep NNs for speech recognition, which achieved SOTA performances across multiple datasets relative to alternative models [68].

Convolutional Neural Networks

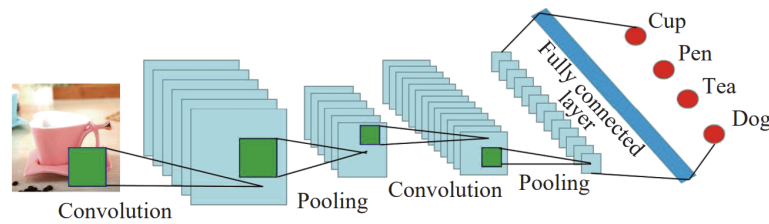


Figure 2.4: A visual representation of a generic convolutional neural network architecture (figure adapted from [156]).

While fully-connected deep NNs were becoming widely adopted across ML domains and tasks, such models were still computationally expensive and the large number of parameters exceeded the hardware constraints of multiple applications. For ordered data, such as text, audio, or images, fully-connected NNs would treat each word, sample or pixel as an independent feature, ignoring any information in the ordered structure of the data. Thus, even though NNs were universal function approximators, many approaches still relied on an initial manual feature extraction step to process high-dimensional inputs into manageable feature vectors. One promising approach to overcome such limitations is to design specialized neurons that leverage inductive bias to significantly reduce the number of parameters.

One of the first classes of algorithms in this direction was the **convolutional neural network (CNN)** in computer vision [90]. Image-based feature engineering approaches often involved the design of multiple 2D digital filters (i.e. kernels) that when convolved with an image, provided new images where pixel values corresponded to features of the local spatial structure at different regions of the image. In CNNs, rather than manually engineering relevant kernels, a kernel is composed of learnable weights, which can be applied to an image input, which followed by the addition of a bias and a non-linear activation function, produces a feature map. Just as multiple neurons are used to produce a hidden feature vector, multiple kernels can be applied in parallel to obtain different image channels of the hidden layer feature map. While this multichannel feature map only contains features regarding local spatial structure, it can be used as the input feature map for a consecutive set of learnable kernels, extracting higher-level features the larger the number of convolutional layers. An example architecture is displayed in Figure 2.4.

One of the most useful properties of CNNs is weight-sharing: the same kernel (often of significantly smaller dimensions than the input image) applies the same weights to different regions of the image. Typically, after multiple convolutional layers, the hidden feature maps are flattened to obtain a feature vector that is then processed by fully-connected layers to obtain the output. To avoid the computational complexity of fully-connected networks, it is common practice to use pooling layers along with convolutional layers [89]. Pooling layers aggregate sub-regions of an image/feature map using operations like the average or maximum, gradually decreasing the dimensions of activations before the fully-connected layers. Such layers also introduce a degree of translation-invariance, since pixel-level activations are often aggregated with neighbouring pixels.

Since the output is still a function of kernel weights, **CNNs** can still be optimized with the standard backpropagation framework. By forcing the network to learn functions that rely on local spatial structure, inductive bias are used to constrain the valid set of learnable functions by the network, reducing the risk of overfitting to the data. Lastly, after training a **CNN** directly on image data, the optimized kernels effectively become the feature extractors that best minimize the loss function, acting as an automatic feature extractor and bypassing the need for domain-specific feature engineering.

While simpler computer vision tasks could be solved with relatively low errors by a wide array of **ML** methods, the introduction of ImageNet, a dataset with over 15 million images of over 22,000 categories provided a challenge that proved to be too complex for feature engineering and traditional classifiers. In 2012, a **CNN** architecture known as AlexNet with approximately 60 million parameters was developed for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), a popular challenge based on an ImageNet subset containing 1000 images of 1000 different classes. AlexNet significantly outperformed the next-best competing methods based on feature engineering and traditional **ML** methods [89]. Specific architectural choices were made to best leverage the use of **GPUs** for faster computation, a strong requirement for a model of this size. The authors also implemented two methods to enforce robustness to real-world image complexities:

- **Data Augmentation:** During training, randomly sampled label-preserving transformations were applied to the images; namely image translations, horizontal reflections, and perturbations in RGB intensities, synthetically augmenting the training set. This forces the model to make predictions with invariance to such transformations, helping it generalize to unseen test set images that may contain images from different angles and perspectives.
- **Dropout:** During the training, the activations of each hidden neuron is set to zero with a probability of 0.5. This method prevents the network from becoming overly reliant on specific feature subsets, ensuring the **CNN** can output correct predictions with a wider variety of features.

This inspired large volume of work into deep **CNNs** for solving complex image-based problems, incrementally improving the capabilities of **CNNs**. For instance, in 2014, the **CNN** architecture from the Visual Geometry Group, referred to as VGGNet, was made significantly deeper than AlexNet (16-19 convolutional layers relative to the 5 used in AlexNet) by decreasing the kernel dimensions, achieving **SOTA** performance in the ILSVRC challenge showing that deeper **CNNs** were able to capture richer representations from images [142].

In 2015, another great contribution was that of Residual learning **CNNs** (ResNets) [66]. Researchers identified that after a certain depth, adding layers contributed to performance degradation relative to having shallower layers without signs of overfitting, suggesting some structural disability. To counteract this, theorising this instability came from the difficulty of later layers learning the identity mapping, residual connections were added to each layer: the activations of the input to a layer are added elementwise to the output. Thus, after enough layers have been used to extract meaningful features, the following layers only need to learn a zero-mapping, which is

arguably easier than the identity. With a model using 34 convolutional layers with residual connections, authors were able to achieve **SOTA** performance in the ILSVRC challenge. Along with other future works, **CNNs** became pervasive across a multitude of computer vision challenges.

While the conception of **CNNs** was in the computer vision domain, their benefits over fully-connected **NNs** are applicable to any ordered data. For instance, 1D **CNNs** (one-dimensional kernels) were applied to raw audio data to achieve **SOTA** performance in music auto-tagging [92], a further example of the capabilities of **CNNs** as automatic feature extractors. Furthermore, in 2017, 1D **CNNs** were also used in sequence-translation tasks, with one implementation achieving **SOTA** performance in the WMT'16 English-Romanian translation task [59], reflecting how with domain-specific architectures, **CNNs** are able to solve highly-complex tasks across domains based on ordered data.

Although the inductive bias of locality in **CNNs** has helped multiple successful architectures solve complex problems in different domains, the same bias can also be a limitation depending on the underlying data structure. While the first hidden feature map pixels only correspond to information processing from a small local region determined by the kernel size, at deeper layers, their values are derived from a wider area of the input image, a concept referred to as the effective receptive field (ERF) or field of view [97]. While the ERF of a **CNN** increases with depth, as well as some other architectural choices, **CNNs** are structurally limited in capturing long-range dependencies, as can often arise in complex document modelling in **NLP**.

Recurrent Neural Networks

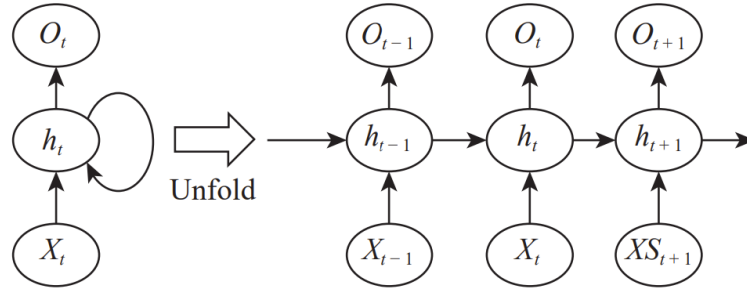


Figure 2.5: A visual representation of a generic recurrent neural network architecture (figure adapted from [156]).

recurrent neural networks (RNNs) can also be used to process a sequential input of the form $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$, where $\mathbf{x}_i \in \mathbb{R}^n$. **RNNs** compute a hidden state feature vector $\mathbf{h}_i \in \mathbb{R}^m$ for each feature vector in the sequence. To leverage the sequential nature of the data, each hidden state is computed as a function of the input feature vector and the hidden state computed for the previous element in the sequence, such that:

$$\mathbf{h}_i = \phi(\mathbf{W}_x^T \mathbf{x}_i + \mathbf{W}_h^T \mathbf{h}_{i-1} + \mathbf{b}) \quad (2.28)$$

where ϕ is the non-linear activation function, $\mathbf{h}_i$ and $\mathbf{h}_{i-1}$ are the hidden state associated with $\mathbf{x}_i$

and $\mathbf{x}_{i-1}$ respectively, $\mathbf{W}_x \in \mathbb{R}^{n \times m}$ is the weight matrix applied to the input vector, $\mathbf{W}_h \in \mathbb{R}^{m \times m}$ is the weight matrix associated with the previous hidden state vector, and $\mathbf{b} \in \mathbb{R}^m$ is the bias added in the computation.

Thus, the hidden state representation of each sequence element is associated recursively with previous sequence elements. As with the previous architectures, the output hidden state feature vectors $\{\mathbf{h}_1, \dots, \mathbf{h}_t\}$ can be used as the input for a subsequent layer to obtain new hidden states, a process that can be carried out multiple times, resulting in deeper [RNNs](#) and more complex representations. As with [CNNs](#), [RNNs](#) uses weight sharing by using the same weight matrices for all sequence elements, using the inductive bias that the context around different sequence elements should be processed in the same manner regardless of where in the sequence specific patterns are observed. Furthermore, this weight sharing also enables [RNNs](#) to efficiently handle variable length sequences, a commonplace challenge in domains such as [NLP](#). To optimize [RNNs](#), backpropagation through time (BPTT) is used, backpropagating gradients from the final state towards the first. Depending on the downstream task, after the final layer, the hidden state of the final sequence element or all hidden states is processed to obtain the desired output. A generic example architecture of an [RNN](#) is displayed in Figure 2.5.

[RNNs](#) have been incrementally improved to address specific limitations. For instance, standard [RNNs](#) are one-directional: only past sequence elements can influence the hidden states of subsequent hidden states. This can often be limiting with data modalities such as text, where for the context of a word, subsequent words are just as important as the previous ones. Thus, bidirectional [RNNs](#) were developed, where independent "forward" and "backward" hidden states are computed from the subsequent and previous sequence elements respectively, with the output of the given sequence element being a concatenation of both states.

Another limitation of [RNNs](#) is that, despite hidden states being recursively dependent on previous states, the contributions of previous states gradually diminishes along the length of the sequence, preventing the model from capturing long-term dependencies and introducing a vanishing gradient problem for earlier layers. Thus, standard [RNNs](#) computations can be replaced with two distinct architectural components: [Long Short-Term Memory units \(LSTMs\)](#) [69] and [Gated Recurrent Units \(GRUs\)](#) [23]. Both these networks contain multiple gating mechanisms that augment the propagation of task-relevant long-term dependencies throughout hidden states. Within the field of [DL](#), [RNNs](#) with [LSTM](#) or [GRU](#) components are often referred to as [LSTMs](#) or [GRUs](#) respectively.

One successful work used [GRUs](#) for machine-translation by *encoding* variable-length sentences into a fixed-length vector representation, followed by *decoding* the fixed-length vector representation into another variable-length sequence to solve an English to French translation task [23]. Bidirectional [LSTMs](#) have also been implemented to learn context-dependent word vector representations, a [self-supervised learning](#) problem that returns meaningful representations of words that can be trained further towards different downstream supervised learning objectives [127]. While [RNNs](#) elegantly handle sequential data and adaptations improved its ability to leverage long-term dependencies, they exhibit one vital limitation. Unlike feed-forward [NNs](#) and [CNNs](#), the sequential computation of hidden states and BPTT during optimization cannot be parallelized,

preventing RNNs from experiencing computational benefits from GPUs.

Transformers

While CNNs excelled at processing ordered data with local structure, they also struggled to capture long-term dependencies. While LSTMs and GRUs are able to capture long-term dependencies in sequential data, their incompatibility with parallelization limits the practical number of parameters and model complexity. Transformers were introduced to resolve both challenges [151].

The original transformer was implemented with distinct encoder and decoder blocks and used for machine translation: generating a target sequence from an input sequence. The encoder unit processes vector representations of each element of the sequence, generating more context-aware representations at each block. In the sequence generation task, the decoder generates the output sequence element at a time (autoregressively). The initial input to the decoder is the output sequence elements generated so far, as well as the context-aware representation of the input sequence generated by the encoder. Through consecutive decoder blocks, the transformer incrementally processes the output sequence representation by extracting relevant content from the input sequence representation. The output of the decoder returns a prediction for the next term in the output sequence.

Rather than CNN or RNN elements, transformers use positional encodings and different variations of attention. Positional encodings are vectors that are added to the initial sequence representation to embed information regarding the location of each sequence element. Attention modules are used to dynamically extract content from the sequence that is relevant to each individual sequence element. The use of positional encodings, attention modules, as well as fully-connected layers, enabled transformers to learn context-aware representations from input sequences, achieving SOTA performances in English to French and English to German translations tasks, with an order of magnitude lower computational complexity during training.

As with other architectures, transformers can be extended to other domains with local grid-like structure. Vision transformers (ViTs) were introduced as an alternative to CNNs by splitting images into a sequence of patches, using their embeddings with an encoder-only architecture for image classification. While for smaller datasets CNNs achieved superior performances, for larger datasets (tens of millions of training examples), ViTs outperformed the SOTA CNNs across different computer vision benchmarks [37]. Thus, when large datasets are provided and sufficient computational resources are available, transformers exhibit strong automatic feature extraction capabilities across a variety of tasks and domains with local grid-like structure.

2.5.7 Surface EMG-Based Deep Learning

DL models have been used to process sEMG for decades, hence are not an inherently new concept. However, without the appropriate inductive biases, fully-connected NNs were too complex for the small dataset sizes typically collected. Thus, most initial DL applications still relied on the same manual feature extraction pipelines, using fully-connected NNs as one of many possible ML models. Furthermore, from the carefully designed features used, traditional ML models performed comparably [5] or exhibited significant performance improvements [40].

As a spatiotemporal signal, there are numerous ways to leverage inductive biases to successfully apply DL methods. The most widely used class of DL algorithms is CNNs, which can be used to process both 1D and 2D ordered data. Arguably the first published implementation of CNNs for sEMG-based ML was applied to the NinaPro gesture recognition benchmark, significantly outperforming SVMs [124]. The input to the CNN were time-windows of recordings from all sEMG electrodes during movement, where the temporal dimension is processed as the sequence, and different electrodes are used as the CNN colour space. By directly processing raw sEMG, this study showed the potential of DL for automatic feature extraction in sEMG ML models. CNN architectures have also been designed to process sEMG time-windows with different electrodes as colour space channels, and while outperformed in gesture classification by random forests with the NinaPro feature set, achieved comparable performances to the general feature sets and classifiers implemented without any feature extraction [4].

While the above CNNs leverage time-dependent inductive biases for automatic feature extraction, treating sEMG sensors as colour space channels does not leverage any inductive biases of the spatial structure of sEMG, which is especially limiting for a large number of electrodes. Thus, another way to use CNNs is to treat regular arrays of multichannel sEMG as instantaneous images, with each electrode corresponding to an image pixel. Images can either be constructed from raw sEMG at every time point (raw images), or from time-window features, which are treated as different colour space channels for each pixel.

From instantaneous images, implemented CNNs significantly outperformed traditional classifiers on multiple publicly available gesture classification datasets (i.e. NinaPro [5], CSL [1] and Capgmyo [38]). Despite the success of either temporal or spatial inductive biases, sEMG is intrinsically a spatiotemporal phenomenon. Thus, 3D CNNs have been implemented, incorporating two spatial dimensions, and the time-axis as a third spatial dimension, which was achieved with 3D convolution kernels [18]. The implemented 3D CNN achieved significantly higher accuracies than spatial CNNs for comparable time-windows.

RNNs have also been used for sEMG gesture recognition. In particular, drawing inspiration from the field of video processing, which is an ordered sequence of images, sEMG signals from an armband were processed first with a 1D CNN layer to capture spatial relationships, followed by LSTM layers to capture sEMG temporal properties, showing strong generalization performance across subjects in a gesture recognition task [83]. The same study also used a hybrid CNN-transformer architecture (conformer), to capture the full spatiotemporal nature of sEMG to predict character handwriting, also exhibiting high intersubject generalization performance. RNNs have also been used to match MU decomposition predictions, exhibiting greater robustness to noise than standard BSS [25].

2.6 The Problem: Generalization & Robustness

2.6.1 Beyond Universal Convergence: The Taxonomy Of Generalization In sEMG Recognition

Numerous ML algorithms have been implemented to decode motor intent from sEMG signals, either through the supervision of discrete movement instructions and joint angle/force measurements, or through the unsupervised extraction of physically meaningful quantities, such as synergies and MU activity. However, most successful studies have been conducted within highly-controlled settings. Within these conditions, even high performance is not a guarantee of translation to practical applications.

As discussed in Section 2.5.1, the aim of ML algorithms is the ability to maximise some task-relevant performance metrics on examples unseen during training, a feat known as generalization. To quantify the generalization of models, a dataset is usually split into a training dataset used for optimizing the model, and a test set to evaluate performance on unseen examples. This approach, however, works under the assumption that the dataset is a representative sample of the data generating distribution. If this assumption is violated, the performance on the test set cannot be used as an indicator of generalization.

Within the context of sEMG-based machine learning, the collected dataset must be representative of the data generating distribution of sEMG signals and features that can be realistically observed during its intended downstream application. In gesture classification tasks, high accuracies have been attained on a test set composed of held-out gesture repetitions from the same recording session [39, 86]. If the data generating distribution represents sEMG signals collected from the same gesture set for the same subject and recording session, reported test accuracies indeed reflect generalization.

However, if the data generating distribution represents sEMG signals from different recording sessions and subjects, or even at different contraction levels or poses, as is often required in real-world settings, the reported performance metrics are not an indicator of generalization. In fact, for gesture classification, numerous studies have developed methods that exhibit high accuracy within a recording session, accompanied by poor performance across different electrode placements [63], recording sessions [60] and subjects [38]. Across these examples, ML models overfit to the data distribution of a single subject and recording session, exhibiting poor performance in unseen subjects and sessions.

The same analysis framework can be applied to MU decomposition. Multiple decomposition algorithms have been applied independently during different contraction levels for single subjects and recording sessions, extracting the activity of numerous MUs [72, 116]. Such linear decomposition algorithms exhibit a high degree of generalization to unseen contractions for the same muscle, contraction level and recording session. This alone has been sufficient to facilitate numerous clinical [119] and neurophysiological findings [149].

However, these algorithms depend on the assumption of a stationarity, as is the case in isometric contractions, and are prone to overfitting when considering other conditions. In dynamic, non-

isometric contractions, BSS methods were shown to only reliably decompose MUs to a speed of 20°/s [118]. One method successfully tracked a large number MUs across recording sessions for the isometric contraction level, but required the marking of the electrode placement on subjects to prevent displacement [100]. Outside of controlled settings, it is often impractical to prevent electrode displacement for every recording session. Thus, while MU decomposition approaches have furthered our understanding of motor control and neurophysiology, current methods are unlikely to generalize to different subjects, recording sessions or even non-isometric tasks, as is often desirable in applications such as the control of prostheses or VR/AR systems.

Ultimately, different levels of generalization are required for specific downstream applications. Model robustness can be extensively defined relative to each confounding factor (e.g. contraction level, electrode displacement, anatomical differences). Practically, three tiers of generalization have been widely investigated:

- **Intrasession Generalization:** For a single subject and recording session, ML models exhibit high performance based on sEMG signals unseen during training from the same subject and session. Models are trained with a subset of the acquired data and tested on held-out sEMG recordings. In this context, high performances have been reported for classification and regression tasks [39, 1, 38, 117], as well as for MU decomposition [72, 116]. This degree of generalization has been leveraged for various motor control and neurophysiological studies, but is insufficient for longitudinal tasks such as the myoelectric control of prosthetics, exoskeletons and VR/AR systems in daily life, as well as MU tracking. For these downstream tasks, models would need new datasets, hence time-consuming data collection, for every new subject or session.
- **Intersession Generalization:** For a single subject, given one or few recording sessions, ML models exhibit high performance based on sEMG signals from the same subject yet a different recording session. Models are trained with a subset of recording sessions, and tested on held-out sEMG signals unseen during training. While numerous methods resulted in poor intersession generalization without any information from the new session (zero-shot) [1, 38], intersession generalization can be substantially improved by adapting the model based on a small subset of sEMG examples from the held-out recording session (few-shot) [1, 38, 27, 44].

While employing experimental constraints to prevent electrode displacement, a substantial number of MUs decomposed from a single recording session have been successfully tracked in a new recording session [100]. Despite the challenges posed by sEMG variability across sessions, intersession generalization enables the longitudinal myoelectric control of external devices from an initial data collection protocol. However, its practicality can be limited by the amount of data required for adaptation for each new recording session.

- **Intersubject Generalization:** After training on sEMG signals acquired from a multiple subjects, ML models exhibit high performance based on sEMG signals from unseen subjects, often referred to as generic models. Generic models are trained with recordings from a subset of subjects, and tested on signals from held-out subjects. This is substantially more

complex than intersession generalization due to the high degree of anatomical differences across subjects, often accompanied by poor performances [1, 60, 38]. However, recent studies have obtained significant improvements across numerous tasks based on DL models trained on data from thousands of subjects [83]. High performances were obtained on completely unseen subjects (zero-shot) and improved further through model personalization based on a small subset of data from new subjects (few-shots). Intersubject generalization is the most practical solution for longitudinal applications. In myoelectric control, this implies that new subjects would be able to directly interact with external devices with little to no data collection required. However, the requirement of initial data collection from thousands of subjects is highly time-consuming and can be unfeasible for downstream healthcare and clinical applications.

Given the limitations of intrasession generalization and the dependence of intersubject generalization on intensive data collection protocols, this thesis focuses primarily on subject-specific intersession generalization. However, achieving robustness in this domain requires isolating the specific physical mechanisms that drive distribution shifts between sessions.

2.6.2 Subject-Specific Intersession Generalization: The Relevance of Electrode Displacement

There are numerous confounding factors that change the sEMG distribution between two recording sessions. While some factors are also present in less controlled settings for the intrasession scenario, all are relevant to prevent overfitting to the sEMG from the motor tasks in the training set. Some of the primary ones are:

- **Electrode Displacement:** Also known as electrode shift, this factor refers to the displacement of electrodes relative to muscle fibre sources between two recording sessions. At the skin surface, a displacement of an electrode significantly impacts the volume conduction effects for underlying active sources (see Section 2.2.2 for more information on volume conduction). Thus, sEMG-based ML models are highly sensitive to any electrode placement deviations between two recording sessions, with a displacement of even 1 cm having significant impacts in generalization performance [63].
- **Contraction Level:** During isometric contractions, the contraction level affects which MUs within a muscle are active and their respective firing rates. Additionally, during dynamic, non-isometric tasks, the contraction of muscle fibres results in a geometric change to muscle anatomy that can significantly affect volume conduction. The former is especially relevant in classification tasks, where a discrete movement can be performed with variable force levels, increasing the within-class variability [16]. While this aspect is naturally captured by regression models, the change in volume conduction during dynamic tasks is also a significant problem. While MU decomposition has been successfully applied to isometric contractions at different force levels, the requirement of stationarity limits its application to fast dynamic contractions [118].
- **Limb Position & Posture:** The position and posture of limbs during movements (or at-

tempted movements) plays a significant role in sEMG fluctuations, as these affect the influence of gravity and biomechanical constraints of movement. Thus, across sEMG-based machine learning, variability in the limb position and posture can result in significant performance detriments [16].

- **Movement Technique:** During sEMG acquisition, users are often unable to perform movements (or attempted movements) with perfect consistency across repetitions. In patients with motor impairments, this may be due to compensatory movements during neurorehabilitation [98]. More generally, in classification tasks, discrete movements can often be generated with different techniques in different repetitions. If the training set is composed of movements that cannot capture these inconsistencies, generalization to different movement techniques can be limited.
- **Powerline Interference:** When using an AC power source, as in common wall outlets, this sinusoidal signal can be radiated to the surroundings, often contaminating low-amplitude sEMG measurements in its vicinity. When unaccounted for, sEMG SNR can be low and models can be sensitive to the specific distribution of powerline interference observed during a single recording session.
- **ECG Interference:** Electrical activity of the heart is a prominent interfering component for sEMG recordings, especially in trunk muscles which is called an electrocardiogram (ECG) artifact. When unaccounted for, models can be sensitive to the specific distribution of ECG activity observed during a single session [113].
- **Movement Artifacts:** When movement is generated at the skin-electrode interface due to the motion of muscles underneath the skin, or when a force impulse travels along the muscle and skin under the electrode, low-frequency movement artifacts are introduced to the acquired signal [31].
- **Baseline Noise Fluctuations:** Additional noise sources originate from the electronics of the amplification system (thermal noise) and at the skin-electrode interface (chemical noise) [31]. Fluctuations in baseline noise result in models overfitting to the noise distribution of the recording session used in the training set.

For intersession generalization, ML models must be invariant to the aforementioned confounding factors. With appropriate sEMG signal preprocessing, some of these factors can be tackled before any feature extraction and modelling. For instance, digital filters such as bandstop filter [38] filters are typically applied to sEMG recordings to reduce contributions from powerline interference, which contains only a single frequency component (at 50 Hz or 60 Hz). Additionally, given their low-frequency content, movements artifacts can be mitigated by using a bandpass low cutoff frequency of 20 Hz [31].

Furthermore, for myoelectric control of external devices, not only can models be trained to capture variability of movements (or attempted movements), but users can also adapt to a specific model during control tasks through experience. In particular, it has been shown in classification-based myoelectric control tasks that subjects with more experience using the controller generated sEMG

signals with greater separability between classes, improving downstream performance [14]. Thus, users can likely minimize sEMG variability due to movement technique, as well as limb position & posture and contraction level through experience, as all these factors can be modulated volitionally.

One key confounding factor that cannot be mitigated through digital filtering or experience is electrode displacement. While subjects can volitionally control force, limb position, posture and general movement technique, electrode displacement is entirely a function of the placement strategy. Therefore, within the context of subject-specific intersession generalization, this thesis primarily focuses on mitigating the effects of electrode displacement.

2.7 Current Solutions For Mitigating The Effects Of Electrode Displacement

From an ML perspective, intersession generalization implies the model exhibits high performance across the true data generating distribution, which includes variability in electrode placement, among other factors that may naturally occur across recording sessions. The challenge is that sEMG signals from a single session are not representative of the true data generating distribution (i.e. all possible recording sessions). Therefore, various methods have been developed to learn meaningful representations that exhibit robust performance on unseen recording sessions. While some were developed to explicitly mitigate the effects of electrode displacement, other methods have focused on mitigating general intersession differences from a statistical adaptation perspective.

2.7.1 Session-Invariant Representations

Large-Scale Pre-training

The most direct way to tackle intersession generalization is through further data collection. The inclusion of multiple recording sessions ensures the training set is a more representative sample of the data generating distribution. While improvements were observed with an increasing number of recording sessions in the training data, performance still remained low when training on four recording sessions. While this can be theoretically improved with a larger number of sessions, collecting dozens of recording sessions per subject is impractical for downstream applications. Thus, large-scale training is limited for intersession generalization and instead, methods must be developed to extract robust representations from one or few recording sessions.

Robust Feature Extraction

In traditional sEMG-based machine learning, feature engineering is used to transform sEMG signals into feature vectors, followed by a model optimized to make predictions based on these extracted examples. To construct feature sets, inductive biases based on domain knowledge are leveraged to obtain representations that are predictive of the quantities of interest for the downstream task [40, 39, 86]. In this context, feature representations may be engineered to be invariant to confounding factors like electrode displacement, where the training dataset is representative of

the data generating distribution within that feature space. However, most features extracted only leverage inductive biases based on temporal dependencies, often being sensitive to electrode displacement. Assessments of feature stability have indicated that despite variations in robustness, classification accuracies remain low ($< 60\%$) under electrode displacement [150].

Therefore, researchers have also investigated feature engineering approaches that leverage spatial inductive biases to reduce the effects of electrode displacement. The variogram, a prominent feature in geostatistics, was introduced to the sEMG-based ML domain. The variogram was estimated as the mean-square of differences between electrode pairs from HD-sEMG arrays separated by a constant spatial lag h . By treating electrodes pairs as different realizations of the same stochastic process, the experimental variogram introduces a location-invariant feature. Across simulated electrode displacements of 1 cm along different orientations, the experimental variogram achieved significantly higher accuracies ($\approx 90\%$ for nine discrete movement classes) than time-domain features using an SVM [144].

Another promising approach applied to HD-sEMG arrays is that of common spatial patterns (CSP), popularized in EEG, which finds a projection of channels that minimises the variance within classes while maximizing the variance between classes [122]. Under simulated electrode displacements of 1 cm, the CSP-based approach significantly outperformed time-domain features and spatial variograms, achieving an accuracy of $\approx 80\%$ for transversal displacements and $\approx 95\%$ for longitudinal displacements for a task with eleven discrete movements.

Despite the promising results in mitigating simulated electrode displacement within the same recording sessions, neither spatial variograms nor CSP have been directly applied to measure true intersession generalization performance. Thus, despite effectively mitigating simulated displacements, such methods may be highly sensitive to other confounding factors in the data. In particular, performance across sessions was shown to degrade for LDA-based classifiers, which relies on similar assumptions to CSP [152]. Therefore, while successfully mitigating electrode displacement, current feature engineering approaches are likely insufficient for robust intersession generalization without some adaptation to data from new recording sessions.

Data Augmentation Strategies

As previously discussed, while in the ideal case the training dataset is a representative sample of the data generating distribution, collecting multiple recording sessions per subject is often an impractical approach to achieved this. However, if the transformations that occur to sEMG signals across recording sessions can be mathematically approximated, the dataset can be synthetically augmented by applying randomly sampled transformations, a framework known as data augmentation. In computer vision, AlexNet [89] popularized the use data augmentation by applying randomly sampled translations to images, increasing the effective training dataset size and improving generalization to pixel displacements [89]. Within the context of sEMG, since image processing techniques can be applied to signals acquired with constant IED, this data augmentation strategy can be directly leveraged to generalize across electrode placements.

For instance, the effect of including virtually translated versions of 2D HD-sEMG recordings in

the training set was investigated in a gesture classification task with seven discrete movements [155]. Data augmentation improved classification accuracy by $\approx 10\%$ on held-out repetitions of the same session with synthetically introduced electrode displacements of up to 1 cm. More recently, synthetic translations have also been used to further augment datasets collected from thousands of subjects to improve intersubject generalization across motor tasks [83]. While data augmentation holds promise, performance improvements in real intersession conditions have not been widely investigated.

2.7.2 Domain Adaptation

When the aim is to extract session-invariant representations, the data generating distribution is assumed to cover all possible sEMG recording sessions. From this premise, feature vectors $\mathbf{x} \in \mathbb{R}^n$ are extracted from a single recording such that the training set is representative of the data generating distribution $P(X, Y)$ across sessions. This is often referred to as zero-shot generalization. Alternatively, distinct recording sessions can be treated as samples from related, but dissimilar data generating distributions, often referred to as domains. Based on a subset of data from a new recording session, the model is adapted to the new domain. This process is known as domain adaptation (DA), and is an example of few-shot generalization [11].

In particular, trained supervised ML models effectively estimate the posterior distribution $P(Y|X)$, also known as the target concept, based on the domain $\mathcal{D} = P(X, Y)$ from which the dataset is sampled. In DA, given a model optimized on a dataset sampled from a source domain $\mathcal{D}_S = P_S(X, Y)$, the aim is to adapt it to a new target domain $\mathcal{D}_T = P_T(X, Y)$ such that the final model is the optimal estimate of $P_T(Y|X)$. Different factors can contribute to the domain shift to be mitigated, and can be categorized as the following:

- **Label Drift:** Changes to the label balance $P(Y)$. While the true target concept $P(Y|X)$ remains constant, the change in label propensity (e.g. number of samples for each joint orientation in a regression task) renders the learned model suboptimal.
- **Covariate Shift:** Changes to the covariate (input) distribution $P(X)$. While the target concept $P(Y|X)$ remains constant, the change in probabilities of feature representations (e.g. number of samples at different contraction levels per movement in classification task) renders the learned model suboptimal.
- **Concept Drift:** While class and covariance distributions remain fixed, the underlying target concept $P(Y|X)$ has changed. Practically, concept drift can be further separated into two sub-categories based on its origins:
 - **Real concept drift:** The physical process behind the data generating distribution has changed. For instance, real concept drift occurs when users perform discrete movements with a different technique, changing the relationship between sEMG signals and a given class.
 - **Virtual concept drift:** While the physical process has not changed, the observations based on this process have changed, resulting in a change in target concept. For exam-

ple, while a user performs discrete movements in the same manner, the sEMG properties can change due to changes in noise properties, electrode placement, effectively changing the target concept.

Since for sEMG-based ML tasks, acquisition protocols are designed to collect balanced datasets, intersession variability can be treated as a concept drift. More specifically, when addressing confounding factors only at the signal acquisition level (e.g., electrode displacement, baseline noise, etc) intersession generalization challenges are framed as realizations of virtual concept drift. The same framework can be applied to the adaptation of unsupervised models by replacing the label Y with latent representation Z without loss of generality.

Model-Based Adaptation

Model-based adaptation approaches work upon the premise that, even if model parameters θ_S^* optimized on source domain data are sub-optimal when applied to data from the target domain, θ_S^* must be a better estimate of θ_T^* than random. Under these conditions, θ_S^* is assumed to be a better initialization than random when determining θ_T^* from a small calibration set. Since model-based adaptation directly updates the estimate of $\hat{P}(Y|X)$, it can be used to mitigate any structure of concept drift.

The effect of model-based adaptation on classification performance across recording sessions was investigated based on the structure of LDA and quadratic discriminant analysis (QDA) [152]. Classifiers were trained with a single session to classify eight hand gestures using sEMG based on a single time-domain feature per channel. For subsequent sessions, one repetition of each gesture was used to update the class means and covariances and held-out repetitions were used as the test set. This approach yielded significant improvements over training classifiers from scratch, achieving $> 90\%$ across all days in healthy subjects. However, for amputees, this adaptation method did not significantly outperform a classifier trained from scratch on the adaptation set.

Since DL models require substantial amounts of data to exhibit high generalization performance, DA enables the use of these models in new domains where sampled data is scarce, as with sEMG recordings. Multiple studies have trained deep NNs on sEMG recordings from multiple subjects, and used DA methods on an adaptation set sampled from the target domain [159, 27, 83]. In particular, these methods used the parameters of this generic model as the initial conditions for gradient-based optimization on the adaptation set, a process known as fine-tuning. This approach resulted in an accuracy improvement of $\approx 50\%$ in a gesture classification task relative to randomly initializing model parameters when optimizing on the adaptation set [159]. However, this method faces a scalability issue: to prevent the model from becoming biased toward specific regions of the output space, the adaptation set must densely cover the target distribution (e.g., at least one repetition of every gesture). This calibration burden grows significantly with the complexity of the task, becoming impractical for systems requiring high-dimensional control.

Since acquiring a fully labelled adaptation set can be cumbersome, an alternative approach is unsupervised model-based adaptation. One deep-learning specific approach applied to sEMG gesture recognition is known as adaptive batch normalization (AdaBatch), where adaptation is performed

by updating the layer normalization statistics using unlabelled target domain data [38]. While accuracy improved by 15% when using AdaBatch, performance remained low ($< 70\%$) while considering practical system response times. Another unsupervised technique known as domain adversarial training was applied to sEMG signals. In this context, domain-adversarial neural networks (DANNs) are trained by extending the network to have two final outputs: the label to predict from the input, and an adversarial classification task, which aims to classify examples as originating from either the source or target domains. However, the performance of the domain classifier is minimized, such that the final network representation is maximally domain invariant, an objective known as adversarial training. When adapting to all available data in the new recording session, significant improvements were observed relative to no adaptation or the use of AdaBatch [26].

However, performance still remained below that reported from supervised model-based adaptation studies. This is consistent with previous research that showed that supervised adaptation significantly outperformed unsupervised methods [140]. Crucially, unsupervised adaptation does not eliminate the requirement for an adaptation set with balanced label distributions. If the user's behaviour during the adaptation set is biased towards specific movements (e.g., primarily flexing the wrist), the adaptation updates will skew the model parameters toward those regions, potentially degrading performance on the unobserved regions of the output space. Thus, model-based adaptation methods that do not leverage inductive biases to mitigate concept drift are effective, but may be limited by the required amount of calibration data.

Input-Based Adaptation

Model-based adaptation aims to mitigate concept drift by adapting the estimate $\hat{P}(Y|X)$, which implicitly assumes that fundamental relationship between examples and labels (or latents) has changed. In the case virtual concept drift, the problem can be framed as inverting the effects of intersession changes at the input level while the model remains static, a process that can be referred to as input-based adaptation.

Let f_θ be a function parametrized by parameters θ optimized on the training dataset sampled from the source domain, such that $D_{train} \sim \mathcal{D}_S$. Under the assumption that the effects of electrode displacement preserve the underlying signal topology (i.e., is bijective), its effects can be modelled with a deterministic function, h , such that $P_S(h(X), Y) = P_T(X, Y)$. From this premise, input-based adaptation is framed as optimizing the parameters ϕ of the function $g_\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ on an adaptation set from the target domain while parameters θ_S^* optimized on the training set remains static. This process generates an induced distribution $P_T(g_\phi(X), Y)$ that approximates the source domain, such that:

$$P_T(g_\phi(X), Y) \approx P_T(h^{-1}(X), Y) = P_S(X, Y) \quad (2.29)$$

Thus, input-based adaptation transforms the target examples such that the source domain optimal model is optimal in the target domain. Most importantly, focusing on input-based instead of model-based adaptation, inductive biases based on extensive understanding of the physical origins of intersession variability can be leveraged. In particular, when using an sEMG electrode

array with constant **IED**, the effects of electrode displacement can be modelled with virtual spatial transformations. Well-established in computer vision [78], virtual translations can be applied to 1D arrays, as well as rotations, scaling and shear to 2D arrays.

Amma et al., (2015) investigated the use of a spatial input-based adaptation for a hand gesture classification task with 2D **HD-sEMG** arrays placed along the circumference of the forearm [1]. Furthermore, this study assumed that the key contributor to intersession variability was a transversal translation of the array (along the forearm circumference), as these translations were shown to cause significantly larger detriments in performance relative to longitudinal displacements [63]. Authors employed two unsupervised image-based methods to estimate specific anatomical landmarks from **sEMG** features, and estimate the optimal translation as the distance between landmarks identified in different sessions. When using three repetitions of a single gesture from a set of 27 classes as the adaptation set, intersession classification accuracy improved by $\approx 15\%$.

Another unsupervised input-based adaptation method determined the transversal and longitudinal translations that minimized the distance between time-domain features across all channels [160]. Thus, input-based adaptation methods that explicitly model spatial transformations were able to improve intersession generalization, in some cases even with a subset of the complete label distribution. However, current spatial transformation approaches often leverage inductive biases that are acquisition set-up specific, and only leverage input **sEMG** features, neglecting relationships learned by the model.

The above methods modelled the intersession changes with a function g_ϕ constrained to the space of affine transformations, leveraging inductive biases of the effects of electrode displacement. However, different function classes can be chosen depending on the desired outcome. One approach introduced mitigates intersession changes with g_ϕ representing a channel-wise linear transformation [85]. For validation, an **LSTM** gesture classifier (f_θ) was trained based on time-windows from **sEMG** arrays. For new recording sessions, parameters θ_S^* were frozen, and a channel-wise linear layer at the input was optimized using an adaptation set. By optimizing an input transformation jointly with the static mapping optimized on the training set, the adaptation procedure implicitly leverages the complex temporal patterns learned by the **LSTM** model. When modelling every transformed channel as a linear mixture of all channels, g_ϕ can implicitly represent spatial transformations (with bilinear interpolation) and impedance changes based on **sEMG** amplitude scaling. This adaptation method improved intersession accuracy by $\approx 37\%$ relative to no adaptation. A linear transformation has more capacity than a few spatial transformation parameters, hence can mitigate a greater extent of intersession confounding factors. However, the increased model capacity of linear layers increases the risk of overfitting, likely requiring a representative sample from the complete label distribution, experiencing similar challenges to model-based adaptation methods.

2.7.3 Motor Unit Tracking

From a probabilistic perspective, the convolutive mixture model assumes the conditional distribution $P(X|Z)$, where Z is a random variable corresponding to **MU** activity. Thus, **MU** decomposition provides an estimate of the posterior distribution $P(Z|X)$. When **MU** decomposition is applied

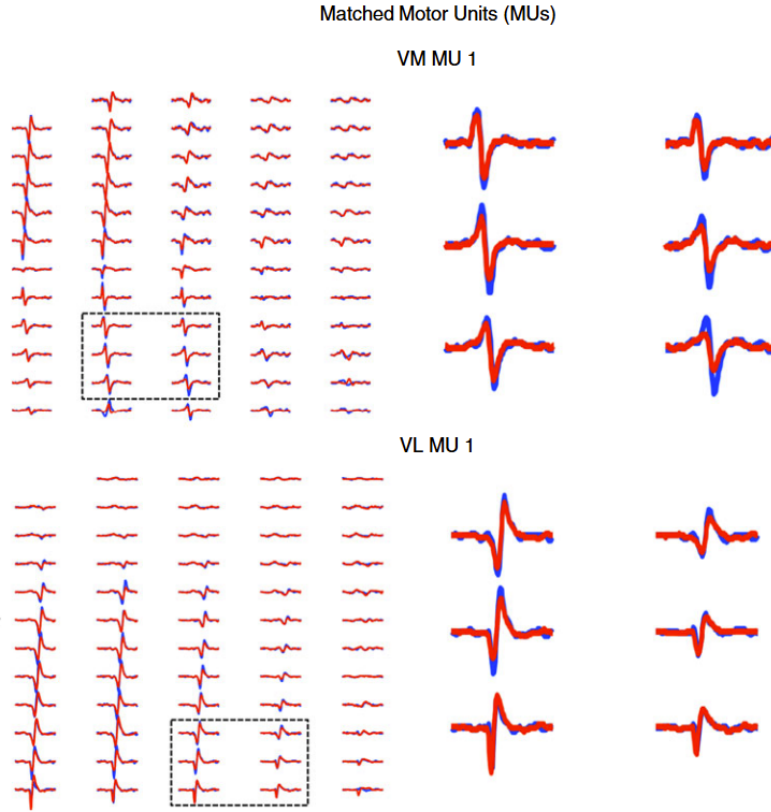


Figure 2.6: Visualization of MUAP distributions of tracked MUs from the intersession MU tracking study of Martinez-Valdes et al., (2017). Results were obtained from HD-sEMG recordings from the vastus lateralis (VL) and the vastus medialis (VM). As shown, for a fixed electrode placement, MUs maintain relatively consistent MUAP distributions across recording sessions (figured adapted from [100]).

to multichannel sEMG recordings, the estimate $\hat{P}(Z|X)$ is obtained for a given contraction level and recording session. Particularly, for convolutive BSS, linear convolutive filters are obtained for each MU that are fully dependent the volume conduction effects between every MU and channel pair. MU decomposition extracts a large number of MUs with high rates of agreement within a recording session and contraction level, exhibiting high levels of intrasession generalization [72, 116]. On the other hand, confounding factors across sessions changes volume conduction effects, rendering extracted filters as sub-optimal. Under the assumption that the physical sources (i.e. MUs) do not significantly change across sessions, decomposition methods can be applied independently for different sessions, having a high probability of decomposing a common subset of MUs. However, given the unsupervised nature of decomposition, this approach loses the one-to-one correspondence between MUs between recordings, limiting its use in applications that require the longitudinal decoding of the same MUs.

Traditionally, MU tracking refers to the matching of MUs independently decomposed across different conditions. For isometric contractions of different contraction levels within the same recording session, volume conduction effects are assumed to remain the same, and sEMG differences are primarily reflected by the active MUs and their respective firing rates. In this context, MUs have been successfully tracked by comparing the MUAP distributions between MUs across contraction

levels [7]. Martinez-Valdes et al., (2017) applied this approach for tracking **MUs** across recording sessions. By preventing electrode displacement with markings on the skin, the comparison of **MUAP** distributions enabled the successful tracking of **MUs** across sessions [100], reflected by the consistent **MUAP** distributions per **MU** observed across sessions. This suggests that mitigating the effects of electrode displacement on decomposition would enable the tracking of **MUs** across sessions under more realistic conditions.

By framing **MU** tracking across sessions as an intersession generalization problem, solutions proposed in other **sEMG**-based **ML** tasks can be considered. Given the current dependence of decomposition on linear approaches, it is likely unfeasible to learn session-invariant filters. Furthermore, as with other tasks, the challenges in intersession tracking can be framed as virtual concept drift, as **MU** properties at every location on the skin surface are expected to be approximately the same across sessions. In this formulation, tracking can be reframed as adapting filters optimized on source domain, such that they are optimal on the target domain. This suggests the use of input-based adaptation methods, which in this context, would imply transforming the input **sEMG** signals such that the original convolutive filters are optimal in the new recording session.

The ability to consistently track **MUs** across recording sessions would enable valuable clinical applications by providing a window into neuromuscular function over time. This methodology could, for instance, be used for single motor unit biofeedback control in **SCI** patients [119], as well as the longitudinal monitoring of disease progression in neuromuscular conditions such as amyotrophic lateral sclerosis (ALS) [153].

2.8 Summary & Research Rationale

Within the context of **sEMG**-based **ML**, while the majority of studies have focused on intrasession generalization, this condition is limited in various myoelectric control tasks within realistic conditions. While intersubject generalization is the most desirable solution, the only study that has exhibited zero-shot intersubject generalization required a time-intensive data collection protocol involving thousands of subjects [83], an approach that may be limiting for downstream clinical and healthcare applications. Furthermore, this study also showed that intersubject generalization could still be substantially improved upon using a personalization procedure, requiring subject-specific data collection from new recording sessions. Thus, the primary focus of this thesis is to investigate methods to ensure practical intersession generalization of subject-specific models.

To address this, I first investigate whether robustness can be achieved solely through data-centric methods, specifically spatial data augmentation. However, recognizing the limitations of purely static augmentation strategies, I subsequently hypothesize that active adaptation is required. I therefore propose an input-based **DA** framework that explicitly models electrode displacement, aiming to optimally transform inputs using only a sparse subset of calibration data. Finally, to demonstrate that this physical inductive bias is not task-specific, I extend this input-based adaptation framework to the unsupervised domain of **MU** tracking, validating its utility beyond standard gesture classification.

Chapter 3

Invariance Learning: Tackling Electrode Displacement With HD-EMG Electrode Subsets

3.1 Introduction

Surface electromyography (sEMG) measures muscular activity from the skin-surface. Thus, it has been extensively explored as a non-invasive method to decode movement intention for interfacing with virtual environments [43], neural prostheses [55] and exoskeletons [110]. Traditionally, the sEMG control problem is modelled as a *gesture*¹ classification or joint-angle regression based on short sEMG intervals [156]. Despite its prowess in laboratory conditions, sEMG-based machine learning methods for myoelectric control have not yet been widely adopted commercially due to limited robustness to sources of interference encountered in day-to-day use.

sEMG signals have multiple sources of variability, such as anatomical differences among subjects, non-stationarities due to force variations and limb position [16], as well as electrode placement. This prevents not only generalization across subjects, but also across sessions for a single subject without repeated calibrations [38]. In particular, electrode displacement, which refers to displacement in electrode position across recording sessions, results in significant drops in performance for even small distances. A 1 cm displacement in a simple task with 10 gestures can result in an average accuracy drop of 25% [63]. Interestingly, this drop in performance was significantly reduced by introducing data from different electrode placements in the training distribution.

While deep learning has shown remarkable prowess in learning invariances to such perturbations, it relies on the existence of enough data to cover the input distribution with fidelity. Unlike other domains, such as computer vision or natural language processing, where deep learning with large datasets has led to significant advancements, sEMG data collection requires time-consuming ses-

¹Throughout this thesis, the standard term *gesture* is used as a macro-level class label for configurations of sustained isometric muscle effort rather than physical movement. In this context, these efforts are captured by the isometric data profiles from the Capgmyo dataset. See the front-matter Glossary for the complete methodological definition.

sions, expensive equipment and trained personnel, making available data scarce. In practice, either simpler models are investigated with adaptive training [38], few-shot learning [38], or traditional feature extraction pipelines are employed [39, 86, 5, 128].

Deep learning has also been used to leverage **high-density surface electromyography (HD-sEMG)** grids, which provide **sEMG** at higher dimensionality and spatial resolution [38, 44]. However, despite the useful information, the data variability is even higher due to its high-dimensionality. Other methods to mitigate the effects of electrode displacement have been explored, such as adopting a more robust electrode configuration [158], borrowing from **convolutional neural networks (CNNs)** in computer vision to learn **HD-sEMG** translation invariance [155], as well as including signals from different electrode placements in the training dataset [64]. Training on multiple **HD-sEMG** channel subsets has been explored as a performance comparison to position verification [65], and treating different **HD-sEMG** grids as a set of simultaneously acquired low-dimensional inputs has been explored in gait cycle prediction [82].

Therefore, I propose training on subsets of **HD-sEMG** channels as an end-to-end solution for gesture recognition and evaluate its effect on intersession performance. The concept is to enable translation from robust algorithms trained on **HD-sEMG** datasets to simpler, low-density EMG systems, such as set-ups used in [5]. The findings suggest that using the remaining data for data augmentation increases the number of samples per trainable **parameter**, with a resulting gesture classifier that is more resistant to electrode displacement. However, despite significant improvements, final intersession accuracy was low relative to previously defined success criteria. In summary:

1. I show that by including subsets of channels at different placements during training (i.e. electrode displacement data augmentation), classification algorithms become less sensitive to electrode displacement, congruent with past literature [63, 64].
2. This augmentation approach results in higher intersession accuracy than training on all available channels at a lower computational complexity.
3. Despite improvements, spatial data augmentation resulted in low intersession accuracy ($< 45\%$) relative to the success criterion defined in Chapter 1 (accuracy $\geq 85\%$).

The proposed method was evaluated using the publicly available Capgmyo dataset [38].

3.2 Methods

3.2.1 Dataset

In this study, experiments were carried out using the raw **HD-sEMG** data from the Capgmyo dataset [38], which recorded signals via a differential, silver, wet electrode array as shown in Figure 3.1, resulting in 128 simultaneous channel recordings acquired at 1000 Hz [154]. As shown in Figure 3.1, eight **HD-sEMG** sub-grids were placed with an untargeted placement strategy, spaced uniformly along the circumference of the forearm. The dataset is split into 3 subsets. Experiment I used data from subset DB_a , containing single sessions from 18 subjects, whereas Experiment II

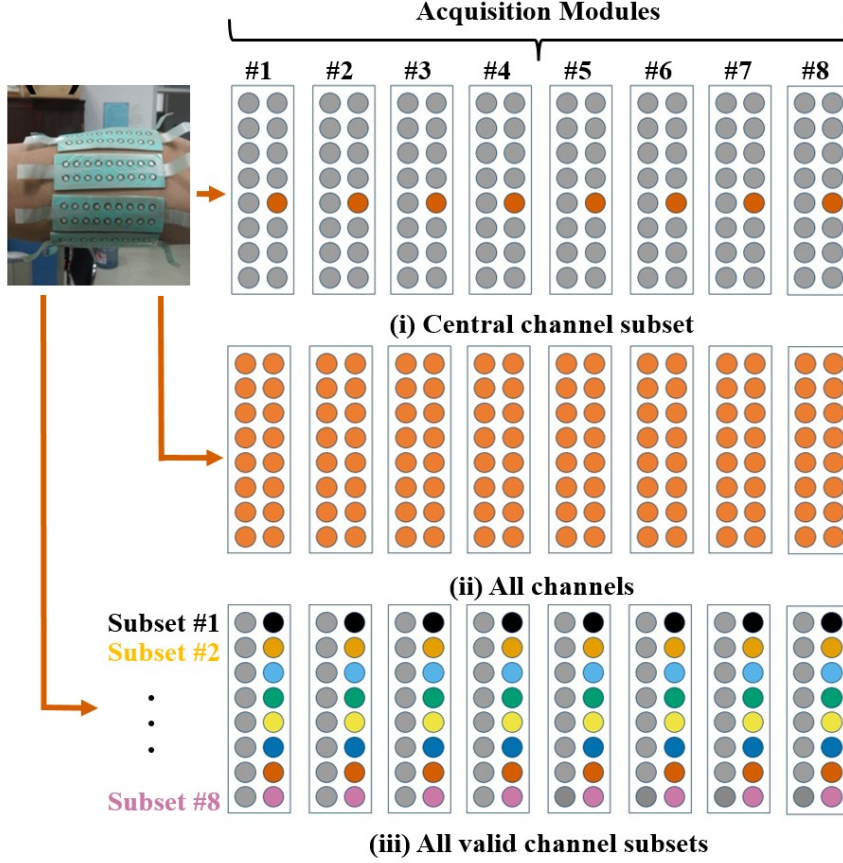


Figure 3.1: Schematic of channel subset sampling procedure for spatial data augmentation applied to the Capgmyo dataset set-up. Three training conditions are considered: **(i)** training only on the central channel subset, where one channel per acquisition channel is considered; **(ii)** training/testing on all (128) channels; **(iii)** training on all valid channel subsets, where only longitudinal translations are considered.

used data from DB_b , containing two sessions for each of the 10 subjects. Due to data corruption in the last subject’s recordings, this experiment was carried forward with the first 9 subjects.

Each session consisted of 10 repetitions of 8 isotonic and isometric gestures ($G = 8$): **(i)** raising thumb upwards, **(ii)** middle/index finger extension with flexion of remaining fingers, **(iii)** flexion of ring and little finger with extension of remaining fingers, **(iv)** thumb opposing base of little finger, **(v)** abduction of all fingers, **(vi)** finger flexion in a fist, **(vii)** pointing index and **(viii)** adduction of extended fingers. Each gesture was held for 3-10s.

Due to the reaction time of the subjects after receiving instructions, acquired labels may not perfectly match gestures. Therefore, as in [38], only the central 1s interval of each gesture was considered in this study. Time-window and stride durations of 256ms ($T = 256$) and 15ms were used respectively. For Experiment I, windows from half of the repetitions of each gesture were assigned to the training set and the remaining half to the test set, resulting in a train/test size of 1960 instances. For Experiment II, the train/test sets were each of the two sessions of a subject, resulting in train/test sizes of 3920 instances.

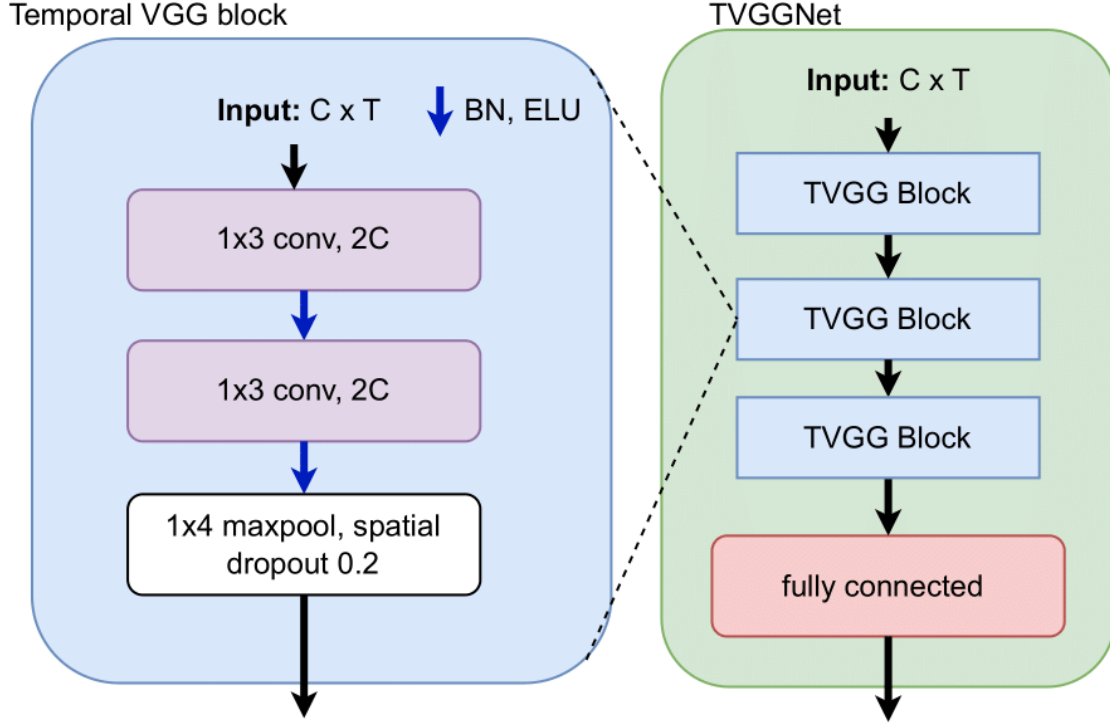


Figure 3.2: The temporal VGGNet variant architecture. The network consists of a number of TVGG blocks consisting of two temporal convolutions, a maxpool and spatial dropout. Each TVGG block doubles the number of channels, while keeping the temporal dimension constant.

3.2.2 Electrode Subset Sampling

Gesture classification is a supervised learning task, with the aim of optimizing parameters θ of a classification model defined as $f_{\theta} : \mathbb{R}^n \rightarrow \{1, \dots, G\}$ that maps input sEMG examples, represented as feature vectors $\mathbf{x} \in \mathbb{R}^n$ into G discrete gestures. This optimization is performed given a dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|D|}$, assuming D samples are independent and identically distributed (iid) and representative of the underlying data generating distribution. Since sEMG datasets are often sparse, the aim was to learn a function f_{θ} that is not only optimal in a single recording session, but exhibits high generalization performance across sessions.

Therefore, I introduced the use of electrode subset sampling as a form of augmentation, with the trade-off of increased effective training set at the cost of the number of input channels, displayed in Figure 3.1. For the Capgmyo data collection protocol, HD-sEMG arrays of 8 rows and 16 columns were used to obtain high-density coverage of the forearm. In this study, instead of using sEMG signals from all channels as inputs, I treated 8 evenly spaced electrode subsets from each row as different lower dimensional recording sessions. While using data from a single recording session, the training dataset used mimicked a low-dimensional training set collected at 8 different locations such as those used in [5].

Since only different rows were considered, classification models were probed to learn representations invariant to longitudinal electrode displacements. Relative to the central channel subset, given the grid geometry, this provided a maximal displacement of 2.4 cm distally and 3.2 cm prox-

imally [154]. Transversal displacements relative to muscle fibre orientation were not included in this study. By treating electrode displacement as a function $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$, this method regularizes model parameters θ such that an electrode displacement invariant mapping is obtained $f_\theta(h(\mathbf{x})) = f_\theta(\mathbf{x})$.

3.2.3 Preprocessing & Machine Learning

Power-line interference was reduced using a digital bandstop filter (45-55H Hz, second-order Butterworth) and all channels were standardized across gestures. As evident from the literature [38], conventional deep learning approaches exhibit limited zero-shot intersession generalization capabilities. This is usually mitigated with calibration frameworks such as [domain adaptation \(DA\)](#) [38, 27]. The primary focus of this study was to enhance intersession generalization of different machine learning models without relying on data from the target recording session. Thus, I considered a temporal [CNN](#) architecture, as well as traditional feature sets commonly applied in [sEMG](#)-based machine learning tasks.

I implemented a [CNN](#) inspired by the VGG architectures extensively used in the computer vision domain [142], with a detailed description provided in Figure 3.2. The architecture used, named [temporal VGGNet \(TVGGNet\)](#), uses 1D temporal convolutions for [sEMG](#) time-windows, treating signals from different electrodes as colour space channels. The primary objective was to assess the effectiveness of the electrode subset sampling method across various machine learning tools rather than pursuing [state-of-the-art \(SOTA\)](#) results. To achieve this broad applicability, I opted for this popular [CNN](#) architecture, upon which many more complex models have been built. Furthermore, the experiments in this study required a network that can process different input sizes, which is something this versatile architecture can perform well [142].

Four traditional [SOTA](#) feature sets were also implemented: Hudgin’s [time-domain \(TD\)](#) features [39], the [enhanced time-domain \(ETD\)](#) features from [86], the best performing features from the original NinaPro dataset publication [5], and the SampEn feature set, which was shown to be the feature set with most stable performance across days [128]. Classification was carried out via [linear discriminant analysis \(LDA\)](#).

3.3 Experimental Analysis

3.3.1 Experiment I: Electrode Displacement Generalization

The first line of inquiry was to investigate whether the feature sets and network architectures considered can generalize to longitudinal displacements with the inclusion of such displacements during training. Training and testing sets were created with a 50:50 split by including every other gesture repetition in each respective set.

For all feature sets (and [TVGGNet](#)), four conditions were considered, where methods could be trained and/or tested on either data from the central subset ([CS](#)) or from all valid subsets ([AVS](#)):

1. ([CS-CS](#)): training/testing on the central subset for baseline performance

Table 3.1: Intrasection test accuracy across Experiment I settings and feature sets (and TVGGNet), averaged across subjects in DB_a.

	CS-CS	AVS-AVS	AVS-CS	CS-AVS
TD	0.920 ± 0.050	0.867 ± 0.047	0.896 ± 0.058	0.659 ± 0.104
ETD	0.944 ± 0.036	0.920 ± 0.033	0.930 ± 0.045	0.672 ± 0.104
NinaPro	0.909 ± 0.057	0.869 ± 0.050	0.891 ± 0.061	0.661 ± 0.098
SampEn	0.950 ± 0.043	0.904 ± 0.046	0.915 ± 0.053	0.646 ± 0.133
TVGGNet	0.861 ± 0.055	0.837 ± 0.060	0.873 ± 0.055	0.706 ± 0.075
Avg. Score	0.917 ± 0.057	0.879 ± 0.055	0.901 ± 0.059	0.667 ± 0.104

2. **(AVS-AVS)**: training/testing on all valid subsets to validate whether models are expressive enough to capture full augmented distribution
3. **(AVS-CS)**: training on all valid subsets and testing on central subset to evaluate whether performance drops at specific locations
4. **(CS-AVS)**: training on central subset and testing on all valid subsets to evaluate the effect of electrode displacement on performance in a usual experimental set-up

The test performance averaged across subjects for each condition is shown in Table 3.1. A one-way ANOVA test indicated that the implemented feature sets and the TVGGNet do not exhibit significantly different test accuracies ($F(3, 284) = 1.640, p = 0.164$). Thus, the average score across feature sets and TVGGNet was considered in this analysis. Given that a Levene test did not support the assumption of homogeneity of variances across experiment conditions ($F(3, 68) = 5.654, p = 0.002$), non-parametric tests were used to compare test accuracies across experimental conditions. Wilcoxon signed-rank tests exhibited significant differences when comparing experimental conditions **(CS-CS)** and **(AVS-AVS)** ($V = 171, p < 0.001$), **(CS-CS)** and **(AVS-CS)** ($V = 156, p < 0.001$) and **(CS-CS)** and **(CS-AVS)** ($V = 171, p < 0.001$). However, it is important to note that while average test performance only reduced relative to **(CS-CS)** by 4% and 2% for **(AVS-AVS)** and **(AVS-CS)** respectively, there is a 25% drop for **(CS-AVS)**. This suggests that training on all valid subsets has a regularizing effect across models, where marginal drops in performance are seen in controlled conditions and far larger improvements are seen in conditions under electrode displacement.

3.3.2 Experiment II: Intersession Performance

Given that electrode displacements can result in significant performance degradation across sessions [157], the effect of my electrode displacement generalization method was evaluated on intersession performance. For each subject, data from one session was added to the training set and data from the remaining session was added to the testing set. Three conditions were considered:

1. **(AVS)**: training on all valid channel subsets and testing on central channel subset

Table 3.2: Intersession test accuracy across Experiment II settings and feature sets (and TVGGNet), averaged across subjects in DB_b.

	AVS	CS	AC
TD	0.407 $\pm$ 0.199	0.343 $\pm$ 0.142	0.198 $\pm$ 0.108
TD (PCA)	0.416 $\pm$ 0.198	0.393 $\pm$ 0.132	0.368 $\pm$ 0.219
ETD	0.425 $\pm$ 0.211	0.363 $\pm$ 0.162	0.226 $\pm$ 0.115
ETD (PCA)	0.437 $\pm$ 0.209	0.367 $\pm$ 0.153	0.414 $\pm$ 0.215
NinaPro	0.407 $\pm$ 0.205	0.335 $\pm$ 0.158	0.205 $\pm$ 0.105
NinaPro (PCA)	0.425 $\pm$ 0.170	0.394 $\pm$ 0.125	0.396 $\pm$ 0.216
SampEn	0.448 $\pm$ 0.217	0.369 $\pm$ 0.173	0.229 $\pm$ 0.111
SampEn (PCA)	0.412 $\pm$ 0.204	0.385 $\pm$ 0.163	0.421 $\pm$ 0.220
TVGGNet	0.420 $\pm$ 0.183	0.365 $\pm$ 0.156	0.394 $\pm$ 0.192
Avg. Score	0.421 $\pm$ 0.198	0.355 $\pm$ 0.155	0.250 $\pm$ 0.146
Avg. Score (PCA)	0.422 $\pm$ 0.188	0.381 $\pm$ 0.143	0.399 $\pm$ 0.207

2. (CS): training/testing on central channel subset

3. (AC): training and testing with a model that used all (128) channels as input, as a naive method to leverage HD-sEMG

Given the high input-dimensionality of condition (AC), performance was also evaluated with principal component analysis (PCA) dimensionality reduction by keeping principal components as to retain 95% of explained variance across conditions. As shown in Table 3.2, the use of PCA resulted in higher test performance for all feature sets included and conditions, with the exception of the SampEn feature set under condition (AVS), as its low feature count was sufficiently regularized by training on all valid channels subsets. When considering these feature sets without PCA, given their test accuracies are not significantly different ($F(3, 212) = 0.441, p = 0.724$), the average score across feature sets was considered.

Paired t-tests showed significantly positive mean differences of +6.9% between (AVS) and (CS) ($t(17) = 3.932, p < 0.001$), and of +20.7% between (AVS) and (AC) ($t(17) = 5.438, p < 0.001$). This suggests that without dimensionality reduction, given the overfitting tendency of the large input dimensionality present, training an 8 input channel model with data from all valid subsets results in significantly greater generalization. This is especially the case when comparing with condition (AC), where performance is far lower than in other cases. The following analysis only includes results using PCA, with the exception of TVGGNet, as it already learns its own feature space remapping.

As before, given a one-way ANOVA test suggested that the implemented feature sets and the TVGGNet do not exhibit significantly different test accuracies ($F(3, 284) = 1.640, p = 0.164$), the

average score across feature sets and **TVGGNet** was considered in this analysis. While there was a significantly positive mean difference of +3.7% between **(AVS)** and **(CS)** ($t(17) = 1.929, p = 0.035$), there was no significant mean difference between **(AVS)** and **(AC)** ($t(17) = 0.966, p = 0.174$). However, despite the lack of significance, **(AVS)** maintained performance with 16x less input channels. Furthermore, there was an average improvement of 2.2% even when excluding the use of **PCA** in condition **(AVS)**. Therefore, training across all valid channel subsets results in significant performance improvements even without **PCA**, and performs comparably or better than other conditions with far lower computational complexity.

3.4 Conclusion

In this chapter, I investigate a novel use of **HD-sEMG** for gesture recognition where only 8 channels were used as inputs, and the remaining recorded channels were used for augmenting the training dataset, as opposed to working with high-dimensional inputs. This method allows different feature sets and networks to generalize across different longitudinal electrode placements, reducing overfitting and consequently leading to significant improvements in intersession performance without the need of dimensionality reduction. Given the high variability in intersession performance improvement across subjects, it is likely that in some cases, other confounding factors such as gesture technique, limb position and contraction level, are the largest contributors to performance degradation, occluding the true robustness benefits of the proposed method.

Crucially, while intersession performance improved significantly relative to the baseline, accuracy remained low across models ($< 45\%$), reflecting the limitations of achieving practical performances for real-world use. In particular, relative to the success criteria introduced in Chapter 1, despite operating within a practical response time and requiring no calibration gestures, the classification accuracies reported were substantially lower than the pre-defined target performance of $\geq 85\%$. Therefore, spatial data augmentation based on electrode subset sampling is unlikely to be a complete solution for robust myoelectric control across real-world conditions.

One limitation of this study is that only longitudinal displacements were considered, whereas transversal displacements typically result in larger performance drops [63]. Preliminary experiments showed that training on these displacements did not contribute to generalization, likely due to transversal displacements relative to the muscle fibre having a larger effect on **sEMG** signal properties as opposed to along the muscle fibre length. Thus, one future line of work is to investigate the use of more expressive deep learning models to obtain classifiers that are robust across all realistic electrode displacements. If promising, this may open a new avenue for **HD-sEMG** in gesture recognition: acquiring complex training datasets that can be leveraged to develop robust algorithms for simpler, inexpensive and commercially viable systems.

Chapter 4

Spatial Adaptation Layer: Interpretable Domain Adaptation For High-Density sEMG

4.1 Introduction

Biosignal acquisition has powered a plethora of healthcare applications and wearable devices. One promising sensing modality is [surface electromyography \(sEMG\)](#), which provides non-invasive measurements related to motor intent. Due to their complex structure, machine learning emerged as a powerful framework for processing [sEMG](#) signals. Furthermore, [high-density surface electromyography \(HD-sEMG\)](#) provides [sEMG](#) signals with high spatial resolution, enabling the use of image-based processing [1] and vision-based deep learning approaches [38] for estimating motor intent.

In [sEMG gesture](#)¹ classification, while various methods yield high performance (accuracy $\geq 85\%$) within the same recording session, performance is generally impractically low across sessions [38]. While generic models have obtained remarkably low error across sessions and even subjects [83], this approach required massive-scale data collection from thousands of participants and produces models that only work with their proprietary [sEMG](#) bracelet, making it financially and practically inaccessible to academic researchers and smaller companies. Therefore, there are no paradigms that result in low classification error across recording sessions without prohibitively expensive and time-consuming data acquisition protocols.

Therefore, different methods have been developed to obtain session-invariant models by addressing the physical origins of intersession performance degradation. The displacement of electrodes across sessions, referred to as electrode displacement, is a key contributor to this degradation in performance [63]. Thus, different methods have been developed to obtain models that are invari-

¹Throughout this thesis, the standard term [gesture](#) is used as a macro-level class label for configurations of sustained isometric muscle effort rather than physical movement. Within this chapter, these efforts generate the spatial topologies of root mean square muscle power processed by the network. See the front-matter Glossary for the complete methodological definition.

ant to electrode displacement, seeking to improve intersession generalization of subject-specific models across sessions, such as spatial augmentation through electrode subset sampling, as shown in Chapter 3. Despite intersession accuracy improvements, performance remained functionally low.

When session-invariant representations cannot be learned for sEMG machine learning tasks, the next-best solution is an efficient calibration procedure that aims to adapt previous subject-specific models to new recording sessions with a small amount of data from the new session. Given that electrode displacement is one of the greatest contributors to intersession variability, several past studies have developed image and signal processing methods to mitigate electrode displacement across two recording sessions. Image-based anatomical landmarks detection have also been used for electrode displacement correction [1], yielding significant performance improvements across recording sessions even with a single gesture for calibration. However, such methods depend on inductive biases of the specific acquisition set-up, limiting its versatility.

The use of image-based deep learning object detection and template matching for computationally efficient displacement estimates in the (x,y) plane of the HD-sEMG grid has also been investigated [160]. In this case, while improvements in computational complexity are achieved, the method is dependent on a convolutional neural network (CNN) architecture that has >10 million parameters, lacking versatility to work with other models. Therefore, despite promising results when tackling the physical origins of the performance degradation, even when only using a subset of all gestures, approaches are traditionally dependent on specific model architectures, acquisition set-up geometry, and further complicated by each method’s individual heuristics.

A more versatile subject-specific model calibration framework is the formulation of intersession changes as a domain adaptation (DA) problem. The objective of DA is to adapt parameters optimized using samples from one data generating distribution to examples sampled from another distribution. For sEMG classification, DA is used to align input distributions across recording sessions or even subjects for the same classification task. The majority of sEMG DA research has focused on model-based adaptation, which involves updating previously learned parameters given an adaptation set from the new recording session. Unsupervised approaches such as adaptive batch normalization (AdaBatch), and supervised approaches such as fine-tuning, progressive neural networks (PNNs), and model-agnostic meta learning [44] have been investigated.

One promising approach uses an input-based adaptation method, introducing a two-step DA procedure. First, a recurrent neural network (RNN) is trained with data from one session, then a channel-wise linear layer learns to map inputs from the second session to the input space of the previous session at every time point. The second step is performed by keeping the RNN weights frozen and learning the linear transformation through optimization on an adaptation set [85]. This method simplifies the problem of intersession domain shifts to learning a single linear transformation that can be prepended to any classification model. However, all model-based adaptation and the above two-step procedure, whether supervised or unsupervised, depend on a balanced dataset to prevent overfitting. This requires the acquisition of at least one repetition of every gesture in a dataset, which can become impractical for real-world applications.

To leverage the efficiency of methods that directly mitigate electrode displacement and the versatility of **DA** methods, I introduce the **Spatial Adaptation Layer (SAL)**, with architecture depicted in Figure 4.1. This layer contains only seven learnable weights that parametrize a spatial transformation to the coordinates of an input regular data array. In this study, I show that **SAL** achieves the following desiderata:

1. **Gesture/repetition efficiency:** across the publicly available datasets shown, **SAL** exhibited competitive performance even when the adaptation set uses only half of the gestures available during training, where intersession performance for other well-established **DA** methods implemented performance only worsens due to class-imbalance overfitting.
2. **Low computational/memory overhead:** the **SAL**-adapted logistic regressor exhibited competitive results relative to other classifiers and **DA** methods, despite only using a few thousand parameters overall.
3. **Interpretability:** in a post-hoc analysis, I show that only for the **SAL** parameter responsible for displacements transversal to the forearm length, a positive correlation with **SAL**-based adaptation performance improvements is observed, in line with expectations based on past studies [157], suggesting **SAL** learns physically meaningful parameters despite only depending on **classification loss**.
4. **Versatility:** as in [85], this layer was trained in a two-step **DA** method and can be prepended to the input of any model that takes in regular grids as inputs. Additionally, **SAL** can successfully mitigate the effects of spatial transformations (translations, rotations, scalings and shear) directly from the objective function of choice, making it independent of set-up specific heuristics.

4.2 Related Work

Although a study has implemented **spatial transformer networks (STNs)** in **sEMG** classification to address electrode displacement for subject-specific models, despite one of the modules being referred to as a 'Fine-Tuning' layer, there is no fine-tuning in the transfer learning sense, with their approach being a domain generalization rather than a **DA** method [21]. **SAL** is substantially different from the usual **STN** module. Although the architecture of **SAL** is derived from **STNs**, the key distinction is that while **STNs** learn a localization network to estimate parameters of an affine transformation as a function of the model input [125], **SAL** removes this localization network altogether and treats the spatial parameters as model weights. Through a two-step adaptation procedure like the one in [85], parameters of a single (spatial) transformation are learned to ensure that the input distribution of the second session matches that of the first. Hence, while **STNs** estimate a set of parameters for every sample, **SAL** learns a unique set of parameters for a new session (domain). While the philosophy is similar to the prepended linear layer, I hypothesised that the reparametrization of the domain shift correction would enable performance improvements even when adapting to subset of all classes, a context in which the linear layer approach is likely to overfit.

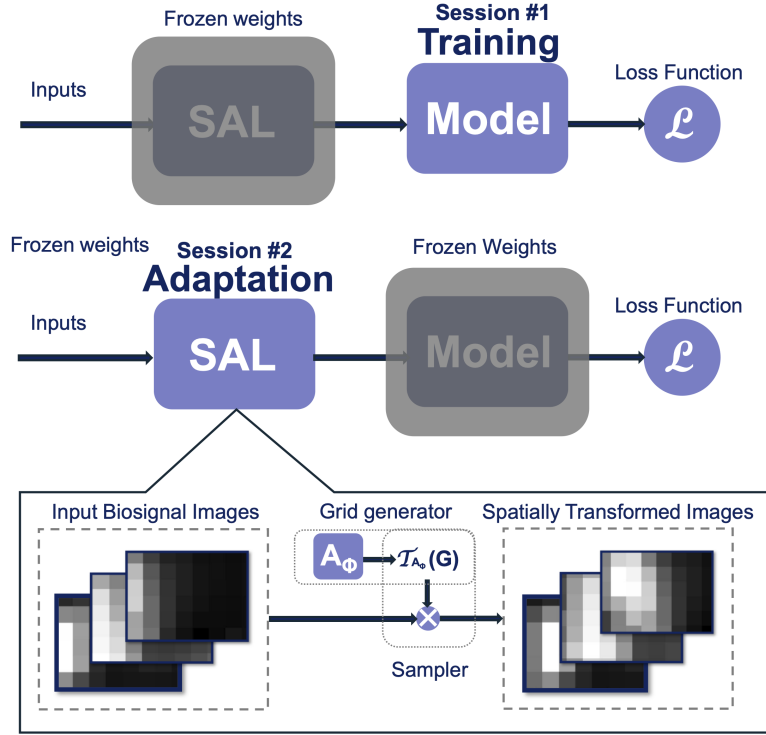


Figure 4.1: Schematic of Spatial Adaptation Layer (SAL) architecture and usage. SAL parameters are frozen and the chosen model is optimized on session #1. On session #2, given a few examples, the model is frozen and SAL parameters are optimized. SAL is similar to a Spatial Transformer module from [78], with the key difference that rather than modelling affine coefficients A_ϕ as a function of the input, ϕ are treated as learnable parameters.

4.3 Methods

4.3.1 sEMG Intersession Calibration: Partial Class Domain Adaptation

For sEMG-based models, by treating two subject-specific recording sessions as being sampled from different data generating distributions, intersession generalization can be framed as adapting a model with parameters θ_S on a sampled dataset from the **source domain** ($\mathcal{D}_S = P_S(X, Y)$) such that it is optimal on examples sampled from the **target domain** ($\mathcal{D}_T = P_T(X, Y)$). Under the assumption of intersession variability being primarily due to differences in signal acquisition, the problem can be treated as that of virtual **concept drift**. This suggests that while the physical process behind sEMG generation has not fundamentally changed, variability at the input-level results in different conditional distributions across domains $P_S(Y|X) \neq P_T(Y|X)$. In DA, parameters θ_S are assumed to be better starting point than training a classifier from scratch on a dataset sampled from $\mathcal{D}_T$ (see Section 2.7.2 for more information on DA). Furthermore, in a classification task, given an adaptation set sampled from the target domain that contains a subset of all available classes ($\mathcal{Y}_{sub} \subset \mathcal{Y}$), I formulated the partial class DA setting. The aim of this procedure was to optimize the model on a dataset sampled from a target domain with a subset of classes the model must generalize for.

Structured Sampling

Unlike classical datasets that assume independent and identically distributed (iid) samples, **sEMG** recordings are collected in **repetitions**. Each repetition r corresponds to a complete movement execution and yields a sequence of N samples. Collecting even a single label-example pair per class requires a full repetition, which motivates the few-gesture calibration setting. Given this sampling procedure, three datasets were constructed from the source and target domains. The number of samples per repetition, N , may vary across repetitions and across datasets (training, adaptation, test).

Training Dataset

The training dataset D_{train} is sampled from the source domain $\mathcal{D}_S$:

$$D_{\text{train}}^{(y)} = \{(x_{r,n}, y)\}_{r=1, n=1}^{R_{\text{train}}, N_{\text{train}}}, \quad D_{\text{train}}^{(y)} \sim \mathcal{D}_S, \quad y \in \mathcal{Y}. \quad (4.1)$$

where R_{train} and N_{train} correspond to the number of repetitions per class in the dataset and the number of samples per respectively. During training, representative samples from all gestures in the classification task are sampled ($\mathcal{Y}$).

Adaptation Dataset

The adaptation dataset D_{adapt} is sampled from the target domain $\mathcal{D}_T$ but restricted to a small subset of classes $\mathcal{Y}_{\text{sub}} \subset \mathcal{Y}$:

$$D_{\text{adapt}}^{(y)} = \{(x_{r,n}, y)\}_{r=1, n=1}^{R_{\text{adapt}}, N_{\text{adapt}}}, \quad D_{\text{adapt}}^{(y)} \sim \mathcal{D}_T, \quad y \in \mathcal{Y}_{\text{sub}}. \quad (4.2)$$

where R_{adapt} and N_{adapt} correspond to the number of repetitions per class in the dataset and the number of samples in a repetition respectively.

Test Dataset

The test dataset D_{test} is also sampled from the target domain $\mathcal{D}_T$ and includes held-out repetitions from all classes $\mathcal{Y}$:

$$D_{\text{test}}^{(y)} = \{(x_{r,n}, y)\}_{r=1, n=1}^{R_{\text{test}}, N_{\text{test}}}, \quad D_{\text{test}}^{(y)} \sim \mathcal{D}_T, \quad y \in \mathcal{Y}. \quad (4.3)$$

where R_{test} and N_{test} correspond to the number of repetitions per class in the dataset and the number of samples in a repetition respectively. It is used to evaluate both adaptation performance on seen classes $\mathcal{Y}_{\text{sub}}$ and generalization to unseen classes $\mathcal{Y} \setminus \mathcal{Y}_{\text{sub}}$.

Partial Class Adaptation Through Inductive Biases

Input-Based DA Framework: Let $f_\theta : \mathbb{R}^n \rightarrow \mathcal{Y}$ be a classifier with parameters θ optimized on the training dataset $D_{\text{train}}^{(y)} \sim \mathcal{D}_S$. Under the assumption that the physical origins of intersession variability can be modelled as a bijective, deterministic function, $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$, such that $P_S(h(X), Y) = P_T(X, Y)$. Thus, to mitigate intersession variability, the function $g_\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ was

introduced, parametrized by ϕ , and applied to examples sampled from $\mathcal{D}_T$. This effectively created the composite predictor $f_\theta \circ g_\phi$. During DA, θ remained static and only ϕ was optimized using $D_{adapt}^{(y)}$ through classification error minimization. Here, I aimed to indirectly enforce:

$$P_T(g_\phi(X), Y) \approx P_T(h^{-1}(X), Y) = P_S(X, Y). \quad (4.4)$$

From this, it can be seen that the relationship $P_T(Y|g_\phi(X)) \approx P_S(Y|X)$ also holds. Intuitively, θ can be seen as responsible for intrasession feature extraction and classification, a problem where numerous studies have reported performances of accuracy $> 95\%$ [1, 38, 27]. Additionally, ϕ is responsible for mitigating intersession variability, separating the challenge of intersession generalization down into two distinct tasks. This has the additional benefit of preventing catastrophic forgetting, as the classifier can always achieve original performance in the source domain by removing the learned transformation g_ϕ . Although the input linear layer previously investigated [85] is one implementation of this input-based DA framework, where ϕ corresponds to the weights and biases of the linear layer, the capacity of this model is high enough to result in overfitting when optimized on a subset of all available classes, as inductive biases were minimally leveraged. Instead, I aimed to construct a function g_ϕ that is less prone to such limitations by leveraging inductive biases regarding the physical origins of variability between sessions.

Generalization Through Inductive Biases: For successful calibration, g_ϕ must be constructed such that optimization of ϕ with classes available in $D_{adapt}^{(y)}$ (i.e. $y \in \mathcal{Y}_{sub}$) must also be able to minimize the loss on all possible classes (i.e. $y \in \mathcal{Y}$), such that:

$$\arg \min_{\phi} \hat{\mathcal{L}}(f_\theta \circ g_\phi; D_{adapt}) \approx \arg \min_{\phi} \mathcal{L}(f_\theta \circ g_\phi; \mathcal{D}_T) \quad (4.5)$$

Beyond assuming that intersession variations can be modelled as a bijective, deterministic function, inductive biases of the physical origins of this variability were leveraged to appropriately constrain the function g_ϕ . Crucially, I neglected confounding factors that can be volitionally controlled by subjects, such as contraction level and limb position [16], under the assumption that subjects can learn to use sEMG-based myoelectric control within a functional operating range.

Therefore, I primarily considered known physical transformations that occur at the signal acquisition level (e.g. electrode displacement [157], skin impedance [130], noise distributions shifts [31], etc). I further assumed this function was completely class agnostic, hence applied equally and independently for every class. Hence, I constrained g_ϕ to bias the transformation to learn the inverse of these physical, underlying transformations, as the optimal physically modelled function for a subset of classes is more likely to also be optimal for unseen classes. Four major physical, class-agnostic transformations were considered to enforce $P_T(g_\phi(X), Y) \approx P_S(X, Y)$:

- **Electrode Displacement:** deviations in electrode positions relative to muscle
- **Powerline Interference:** additive, session-specific, activity-independent sinusoidal component from AC power source per channel

- **Baseline Activity Fluctuations:** additive, white Gaussian noise with session-specific variances per channel
- **Outlier Channels:** dead or corrupted channels that do not represent physiological information.

By assuming that powerline interference, baseline activity fluctuations and outlier channels across sessions are accounted for with appropriate preprocessing, I constructed a function g_ϕ that exclusively focuses on electrode displacement.

Therefore, I introduced an input-based DA method that, with very few learnable parameters, can only capture uniform geometric changes in the sEMG spatial patterns. Since the transformation operates purely on spatial coordinates rather than class-specific features, the mapping that corrects electrode misalignment for a subset of classes should be the same correction for all possible classes. In other words, the limited parameter space limits the transformation from learning class-selective adjustments, with the aim of only learning a single global geometric correction that improves the spatial alignment for all classes. Algorithm 1 outlines the practical implementation of this spatially-oriented input-based DA pipeline.

4.3.2 Root Mean Square: A Physiologically Meaningful Feature

The root mean square (RMS) is a prominent time-domain feature that has been extensively employed in sEMG-based machine learning [86, 1]. Despite its simplicity, as shown in [50], the mean square (MS) signal can be approximated as a function of motor unit (MU) discharge rates and motor unit action potential (MUAP) waveforms. This establishes a direct link between the sEMG signal amplitude and the neural control signals driving muscle contraction, as greater contraction levels increase not only firing rates, but also the number of active motor units and the average MUAP energy. Instead of MS, I chose RMS since it has the same units as the original signal and is a well-established feature in sEMG-based machine learning, while still preserving its monotonic relationship with contraction levels that is useful for classification objectives. For more information on the relationship between RMS and MU properties, see Appendix A.

HD-sEMG Activity Images

Additionally, due to biophysical impedance, the sEMG signal at a given channel is spatially selective and most sensitive to the underlying muscles closest to the electrode, providing localized information. Therefore, by using a two-dimensional HD-sEMG electrode array, the discrete-time representation $x_k^{RAW}[m, n] \in \mathbb{R}^{H \times W}$ was obtained for every time index k and electrode array positions (m, n) , where different electrode locations can efficiently estimate quantities related to the contraction levels of different underlying muscles. These instantaneous images can be aggregated over a time-window to obtain activity images $x_k^{RMS} \in \mathbb{R}^{H \times W}$, defined as the RMS of each channel within the designated time-window centred at index k :

$$x_k^{RMS} = \sqrt{\frac{1}{2T_{RMS} + 1} \sum_{\tau=-T_{RMS}}^{T_{RMS}} (x_{k+\tau}^{RAW})^2} \quad (4.6)$$

To obtain an instantaneous estimate of **RMS** in this offline study, the raw **sEMG** activity was squared, then convolved with a moving average (MA) filter of length $2T_{RMS} + 1$, introducing a delay of T_{RMS} to the system, fixed to an integer number of samples equivalent to 250ms. Since this study only used activity images, for the remainder of this study x_k denotes the activity images centred at sampling index k without ambiguity.

4.3.3 Spatial Adaptation Layer

Layer Formulation

In **SAL**, shown in Fig 4.1, I aimed to parametrize a function g_ϕ that exclusively models spatial transformations. I treated each temporal slice of the given 2D regular array as a grayscale image of size $H \times W$. I used the **sub-differentiable** image sampling operator proposed in **STNs** [78]. Within this framework, the pixels of the output image of size $H' \times W'$ denoted by coordinates (m_i^t, n_i^t) where $\forall i, i \in [0, (H' - 1) \cdot (W' - 1)]$, are mapped to corresponding pixel coordinates in the source image, (m_i^s, n_i^s) , with affine grid mapping A_ϕ , as in (4.7). Transformations are applied to normalized coordinates, where $(\tilde{m}_i^t, \tilde{n}_i^t) = (\frac{2m_i^t}{H'-1} - 1, \frac{2n_i^t}{W'-1} - 1)$, such that $\tilde{m}_i^t, \tilde{n}_i^t \in [-1, 1]$.

$$\begin{pmatrix} \tilde{m}_i^s \\ \tilde{n}_i^s \\ 1 \end{pmatrix} = A_\phi \begin{pmatrix} \tilde{m}_i^t \\ \tilde{n}_i^t \\ 1 \end{pmatrix} \quad (4.7)$$

Using difference source/output image dimensions further enables downsampling/upsampling, which is leveraged by **STN** modules in complex architectures. However, this was not directly useful in the **SAL** formulation and in this study, $H = H', W = W'$ was considered. Once corresponding pixel coordinates for the source image are obtained for all output pixels, the output image was obtained by sampling from the source image, leveraging a sampling kernel w with parameters Φ_m and Φ_n , as in (4.8). The kernel enabled sampling from the source image space continuously despite having a discrete set of source pixels.

$$x'_k[m_i^t, n_i^t] = \sum_{m_j=0}^{H-1} \sum_{n_j=0}^{W-1} x_k[m_j, n_j] w(m_i^s - m_j; \Phi_m) w(n_i^s - n_j; \Phi_n) \quad (4.8)$$

For bilinear interpolation this can be written as:

$$x'_k[m_i^t, n_i^t] = \sum_{m_j=0}^{H-1} \sum_{n_j=0}^{W-1} x_k[m_j, n_j] \max(0, 1 - |m_i^s - m_j|) \max(0, 1 - |n_i^s - n_j|) \quad (4.9)$$

The authors in [78] showed that the partial derivatives of the output image with respect to the input and sampling grid can be defined as:

$$\frac{\partial x'_k[m_i^t, n_i^t]}{\partial x_k[m_j, n_j]} = \sum_{m_j=0}^{H-1} \sum_{n_j=0}^{W-1} \max(0, 1 - |m_i^s - m_j|) \max(0, 1 - |n_i^s - n_j|) \quad (4.10)$$

$$\frac{\partial x'_k[m_i^t, n_i^t]}{\partial n_i^s} = \sum_{m_j=0}^{H-1} \sum_{n_j=0}^{W-1} x_k[m_j, n_j] \max(0, 1 - |m_i^s - n|) \begin{cases} 0, & |n_j - n_i^s| \geq 1 \\ 1, & n_j \geq n_i^s \\ -1, & n_j < n_i^s \end{cases} \quad (4.11)$$

yielding a sub-differentiable sampling mechanism, seen in Eq. 4.11. In this context, backpropagation is still possible since $\frac{\partial n_i^s}{\partial \phi}$ and $\frac{\partial m_i^s}{\partial \phi}$ can be easily derived from Eq. 4.7. For more interpretable parameters to be learned, A_ϕ was further decomposed into subtransforms while preserving differentiability in the operator. This was done by making use of transformations in homogeneous coordinates to obtain the general affine transformation A'_ϕ defined in Equation 4.12:

$$A'_\phi = T \cdot R \cdot S \cdot Sh \quad (4.12)$$

where: $Sh = \begin{pmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad S = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$

$$R = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{pmatrix} \quad (4.13)$$

with T , α , s and sh representing translation, rotation, scaling and shearing respectively.

In this context, the parameters used to mitigate intersession variability are $\phi = [T_x, T_y, \alpha, s_x, s_y, sh_x, sh_y]^T$. Hence, A_ϕ was redefined as the submatrix of A'_ϕ such that $A_\phi = A'_\phi[I, J]$ where $I = \{1, 2\}$ and $J = \{1, 2, 3\}$. Given tensor slicing and multiplication are differentiable operations, gradients are preserved with no loss of information and can be defined with respect to the sampling grid coordinates by applying the chain rule. This affine transformation and layer was parametrised by seven learnable parameters. Ordinarily, one would expect such parameters to be learned through a similarity metric applied to images directly as in image registration. However, exact correspondence may be ambiguous in different scenarios. Instead, parameters can be learned for any optimization based downstream task. Here, classification loss backpropagated for multiple windows during adaptation, guiding the learning of these physically meaningful parameters to uncover a unique mapping from session to session.

Optimization Strategy

The optimization of **SAL** presents distinct challenges due to the non-convex loss landscape created by the seven parameter affine transformation. While some minima represent semantically meaningful transformations that enhance feature discriminability, others may still correspond to alignments that reduce training loss through dataset-specific artifacts rather than generalizable spatial transformations. Poor initialization can thus lead to convergence toward suboptimal alignments.

Hybrid Optimization Strategy: To navigate this optimization landscape effectively, a two-stage approach was employed, combining systematic parameter space exploration with targeted local

optimization. **Latin Hypercube Sampling (LHS)** was applied across a restricted range of transformation parameters, evaluating classification loss left-out adaptation data. Parameter bounds ensure physically plausible transformations while providing sufficient parameter space coverage. The **LHS** approach ensures efficient sampling density across all dimensions, while still capable of exploring regions between grid points. The configuration yielding the lowest classification loss serves as initialization for gradient-based optimization. This approach leverages the insight that high-quality starting points are more likely to lead to meaningful spatial transformations that generalize beyond training data, effectively balancing global exploration with local refinement efficiency.

Soft Parameter Constraints: To prevent instability caused by unbounded parameter updates during gradient-based optimization, soft constraints were applied to the affine transformation parameters using a hyperbolic tangent function. The unconstrained parameters $\hat{p}$ were passed through $\tanh(\cdot)$, producing values in $[-1, 1]$, which were then linearly mapped to predefined parameter-specific ranges $[p_{\min}, p_{\max}]$ as follows:

$$p = 0.5 \cdot (\tanh(\hat{p}) + 1) \cdot (p_{\max} - p_{\min}) + p_{\min}.$$

This formulation ensures that all parameters remain within physically plausible limits while maintaining smooth gradients for optimization. It effectively prevents gradient overflow and excessive distortions, improving training stability without the use of hard clipping. Unless otherwise stated, all reported distances and parameter values correspond to these *constrained* parameters.

Aggregated Activity Images: Since activity images within finite-time windows are a noisy estimate of the quantity of interest, given the central limit theorem, longer time-windows were expected to provide lower noise estimates. Therefore, given the computational demands of evaluating multiple transformation configurations, the parameter search was simplified by computing the **RMS** of each channel across all temporal samples for each class during the adaptation phase. This temporal aggregation created a single, more robust and representative spatial pattern per class, referred to as aggregated activity images, dramatically reducing computational overhead.

4.3.4 Baseline Normalization

Session-Level Assumptions: In [1], under the assumption that baseline activity is constant during a recording session for every channel, their effects on the **RMS** estimates can be mitigated by subtracting the **RMS** during rest. The analysis below shows instead that, under the same assumptions, it is preferable to carry out the analysis at the mean-square level.

Single Channel Analysis

In the following analysis, $x(t)$ was considered to be a continuous-time raw **sEMG** signal for a single channel with no loss of generality, as baseline normalization is performed independently for every channel. By assuming that baseline activity behaves as zero-mean white Gaussian noise, that is, $n(t) \sim \mathcal{N}(0, \sigma_b^2)$, observed **sEMG** can be defined as follows:

Algorithm 1 Selective Target Domain Adaptation for sEMG Intersession Calibration

Require: Training dataset D_{train} sampled from the source domain $\mathcal{S}$ with all classes $\mathcal{Y}$, Adaptation dataset D_{adapt} sampled from the target domain $\mathcal{T}$ with subset $\mathcal{Y}_{sub} \subset \mathcal{Y}$, Training epoch count E_s , Adaptation epoch count E_t , Latin hypercube sample size N

Ensure: Adapted classifier $f_{\theta^*} \circ g_{\phi^*}$ for target session

Phase 1: Source Domain Training

- 1: Freeze spatial transformation parameters ϕ ($f_{\theta} \circ g_{\phi} = f_{\theta}$)
- 2: **for** $e = 1$ to E_s **do**
- 3: Train source domain classifier $f_{\theta} \circ g_{\phi}$ on D_{train} using standard loss $\mathcal{L}_{source}$
- 4: **end for**
- 5: Freeze parameters θ^* of source classifier $f_{\theta^*} \circ g_{\phi}$

Phase 2: Target Domain Data Preprocessing

- 6: **for** each class $y \in \mathcal{Y}_{sub}$ **do**
- 7: Extract all target examples from class y , $\{x_k | (x_k, y) \in D_{adapt}^{(y)}\}$
- 8: Extract one activity image per class: $\bar{x}^{(y)} = \sqrt{\frac{|D_{adapt}^{(y)}|}{\sum_{i=1}^{|D_{adapt}^{(y)}|} x_k^2}} x_k$
- 9: **end for**
- 10: Create aggregated adaptation dataset $\tilde{D}_{adapt}^{(y)} = (\bar{x}^{(y)}, y)$, $y \in \mathcal{Y}_{sub}$

Phase 3: Parameter Space Exploration via Latin Hypercube Sampling

- 11: Generate Latin hypercube sample space $\Phi = \{\phi_1, \phi_2, \dots, \phi_N\}$
- 12: Initialize best parameters ϕ^* and best loss $L^* \leftarrow \infty$
- 13: **for** $j = 1$ to N **do**
- 14: Set transformation parameters $\phi \leftarrow \phi_j$
- 15: Evaluate loss $L_j = \mathcal{L}_{avail}(\phi)$ on $\tilde{D}_{adapt}^{(y)}$
- 16: **if** $L_j < L^*$ **then**
- 17: $\phi_{opt} \leftarrow \phi_j$, $L^* \leftarrow L_j$
- 18: **end if**
- 19: **end for**

Phase 4: Fine-tuning on Target Domain

- 20: Initialize unfrozen transformation parameters $\phi \leftarrow \phi_{opt}$
 - 21: Freeze source domain trained parameters θ
 - 22: **for** $e = 1$ to E_t **do**
 - 23: Fine-tune $f_{\theta}^* \circ g_{\phi}$ by minimizing $\mathcal{L}_{avail}(\phi)$ on $\tilde{D}_{adapt}^{(y)}$
 - 24: **end for**
 - 25: **return** Composite classifier $f_{\theta^*} \circ g_{\phi^*}$
-

$$x_{obs}(t) = x_{true}(t) + n(t) \quad (4.14)$$

RMS Estimation Under Baseline Noise: By assuming $x_{true}(t)$ to also be a zero-mean signal, the **RMS** estimate was defined as follows:

$$RMS_{obs}^2 = \text{Var}\{x_{obs}(t)\}$$

$$RMS_{obs}^2 = \text{Var}\{x_{true}(t)\} + \text{Var}\{n(t)\}$$

$$\text{RMS}_{\text{obs}} = \sqrt{\text{RMS}_{\text{true}}^2 + \sigma_b^2} \neq \text{RMS}_{\text{true}} + \sigma_b \quad (4.15)$$

Key Observation: Thus, while the estimate of baseline activity from rest activity is valid, since $x_{\text{true}}(t) = 0$ is expected during rest, subtracting directly from the **RMS** would not optimally reduce baseline activity.

Power-Level Normalization Approach

Therefore, rather than carrying out baseline normalization at the **RMS** (amplitude) level as in [1], I proposed to perform it at the RMS^2 (power) level, then apply the square root after. For the rest of the evaluations in this study, the former was referred to as **BN-RMS** and the latter as **BN-MS**.

Extension to Powerline Interference

Finally, under assumptions of constant amplitude/phase powerline interference, the same analysis can be used to show that **BN-MS** can reduce its effects. By defining the effect of powerline interference as $p(t) = A \sin(2\pi ft + \phi)$, the following was obtained:

$$x_{\text{obs}}(t) = x_{\text{true}}(t) + n(t) + p(t)$$

$$\text{RMS}_{\text{obs}}^2 \approx \text{RMS}_{\text{true}}^2 + \sigma_b^2 + \frac{A^2}{2} \quad (4.16)$$

Variance of Sinusoidal Interference: where $\text{Var}\{p(t)\} \approx \frac{A^2}{2}$ as long as the time window used to estimate RMS_{obs} is significantly larger than the powerline interference frequency, allowing a sufficient number of complete cycles.

4.3.5 Datasets

CSL

Bipolar recordings were acquired at a sampling rate of $f_s = 2048$ Hz via an electrode array with 192 electrodes and an **interelectrode distance (IED)** of 1 cm [1]. Differencing between consecutive electrodes resulted in a regular grid of $H \times W = 7 \times 24$, totaling $N_{ch} = 168$ channels. This dataset contains recordings from $S = 5$ subjects across $N_{\text{sess}} = 5$ different days. In every recording, subjects performed $R = 10$ repetitions of 27 different gestures. For baseline normalization, the idle gesture repetitions were considered. For practical intersession evaluation, a dataset of six hand gestures ($|\mathcal{Y}| = 6$) was considered. To evaluate effect of the number of available gestures for adaptation, the following gesture subsets were considered:

- $|\mathcal{Y}_{\text{sub}}| = 6$ (all gestures available): thumb flexion, index finger flexion, thumb-index finger pinch, making a fist with force, a stretch with force of all fingers, and extension of thumb, index and middle fingers while keeping ring and pinkie fingers bent.

- $|\mathcal{Y}_{\text{sub}}| = 3$: making a fist with force, a stretch with force of all fingers, and extension of thumb, index and middle fingers while keeping ring and pinkie fingers bent
- $|\mathcal{Y}_{\text{sub}}| = 1$: making a fist with force

Capgmyo

Signals were acquired via eight differential, silver, wet electrodes evenly spaced around the arm, resulting in an irregular $H \times W = 8 \times 16$ grid totaling $N_{ch} = 128$ channels, recorded at a sampling rate of $f_s = 1000$ Hz. Experiments used data subset DBb, containing $N_{\text{sess}} = 2$ recording sessions per subject. Due to data corruption in the final subject's recordings, only $S = 9$ subjects were used. Each session included $R = 10$ repetitions of 8 isotonic and isometric hand/finger gestures, each held for 3–10s. As in [38], only the central $T = 1$ s of each gesture was used. This dataset represents within-subject, intersession data with an irregular HD-sEMG grid. For baseline normalization, rest periods between consecutive repetitions were used. For practical intersession evaluation and comparisons with CSL, a dataset of six hand gestures ($|\mathcal{Y}| = 6$) was considered. To evaluate effect of the number of available gestures for adaptation, the following gesture subsets were considered:

- $|\mathcal{Y}_{\text{sub}}| = 6$ (all gestures available): thumb up, extension of index and middle with flexion of the others, thumb opposing base of little finger, abduction of all fingers, fingers flexed together in a fist and pointing of index finger
- $|\mathcal{Y}_{\text{sub}}| = 3$: extension of index and middle, flexion of the others, abduction of all fingers and fingers flexed together in a fist
- $|\mathcal{Y}_{\text{sub}}| = 1$: fingers flexed together in a fist

4.3.6 Processing

For all datasets used, as in [38], signals were processed with a digital bandpass filter (20-380 Hz, fourth-order Butterworth) and power-line interference was reduced using a digital band-stop filter (45-55 Hz, fourth-order Butterworth). Once activity images were obtained from the raw HD-sEMG, the baseline normalization procedure of choice was performed with recorded segments of rest gesture (idle). Consequently, a spatial median filter was applied to each activity image frame, with the benefits of spatial smoothing and outlier removal. For the CSL dataset, a 3x3 kernel was used, whereas for Capgmyo, which has a variable IED in the transversal direction, a 3x1 kernel was used. Since, for the CSL dataset, gestures were performed for a short time period ($T = 3$ s), to ensure that the labelled data corresponds to the desired movement, the activity segmentation algorithm from [1] was used to determine the active movement segment.

4.3.7 Classification

To demonstrate the flexibility and simplicity of SAL, it was integrated it into two different classifiers. The first is a logistic regressor trained with 15 epochs and batch size of 128. Secondly, the state-of-the-art (SOTA) sEMG image classifier from [38], referred to as CapgmyoNet, was imple-

mented and trained with 2 epochs and a batch size of 128. While 2 epochs may seem like a low amount of training, the stride of 1 when extracting activity images was significantly lower than most previous studies. This implies that for each epoch, the model iterated over a significantly larger number of example images. In the case of the logistic regressor, which is not naturally efficient for processing image data, to prevent overfitting to noisy/corrupted channels during training, dropout at the input ($p = 0.5$) was considered, probing the system to reduce dependencies on a small number of specific channels. All classifiers were trained using the Adam optimizer [87], cross-entropy loss, and a base learning rate of 0.001.

4.3.8 Implementation Details

Model training and evaluation were performed using an NVIDIA L40S GPU (48 GB GDDR6 ECC memory, 18,176 CUDA cores, 568 Tensor Cores, Ada Lovelace architecture). This configuration provided sufficient computational capacity for efficient training of the PyTorch-based **sEMG** classification and **DA** models, even when performing the **LHS** brute-force search. For **SAL** implementation, in-built PyTorch `torch.nn.functional.affine_grid` and `torch.nn.functional.grid_sample` functions developed for the original **STN** implementation were leveraged [78]. The differentiating feature between **SAL** and **STNs** is that **SAL** treats affine transform parameters as learnable parameters, as opposed to the output of a localization network.

4.3.9 Performance Metrics

Each activity image results in one gesture prediction, and metrics are computed over all predictions in the test sets. Accuracy is defined as the ratio of correctly predicted labels to the total number of predictions, given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP , TN , FP , and FN are true positives, true negatives, false positives, and false negatives, respectively. To capture performance in imbalanced scenarios, the f1-score was also reported, defined as the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall},$$

where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. Since performance is mainly used to contrast different methods, the performance difference was defined as:

$$\Delta = (Accuracy^{(1)} - Accuracy^{(2)}, F1^{(1)} - F1^{(2)})$$

where the superscript indexes refer to two distinct sets of adaptation conditions.

4.3.10 Domain Adaptation Methods

To validate performance gains from using **SAL**, beyond interpretability and parameter efficiency, four alternatives (excluding zero-shot performance) were considered. Given a model with parameters θ optimized on the training dataset $D_{train}^{(y)}$, each of these methods leverage a (commonly small)

number of samples from the adaptation dataset $D_{adapt}^{(y)}$. In this study, activity image aggregation was only applied when using SAL-based DA.

Fine-Tuning

The most traditional form of transfer learning and DA, which involves using source domain optimal parameters θ_S^* as an initialization for an optimization procedure on samples from $D_{adapt}^{(y)}$.

Adaptive Batch Normalization (AdaBatch)

An unsupervised DA method popularized in sEMG literature by [38]. All model parameters were frozen, the statistics of all Batch Normalization layers in the given model are updated with samples from $D_{adapt}^{(y)}$. In this study, classifiers were trained on data from a single recording session, hence did not consider the multi-stream extension introduced in [38].

Input Linear Layer

Similarly to how SAL is applied, a linear layer with weight matrix $\mathbf{W}_L \in \mathbb{R}^{N_{ch} \times N_{ch}}$ and bias vector $\mathbf{b}_L \in \mathbb{R}^{N_{ch}}$ was prepended to the classifier [85]. The parameters of $\mathbf{W}_L$ and $\mathbf{b}_L$ were updated with a training procedure using the same learning scheme as the original model training, while the original model parameters were kept frozen. The aim was to learn a linear transformation that transforms $\{x_k | (x_k, y) \in D_{adapt}^{(y)}\}$ that minimizes error of a model originally trained with $D_{train}^{(y)}$, indirectly inverting transformations responsible for the concept drift.

Training From Scratch

Ignoring the training dataset $D_{train}^{(y)}$, classifiers were naively trained with the limited data available from the adaptation dataset $D_{adapt}^{(y)}$.

4.4 Results

4.4.1 Experiment I: Simulated Transformations Performance

To evaluate the capability of SAL to mitigate the effect of spatial transformations between recording sessions of regular biosignal arrays, virtual spatial transformations were applied to held-out data of the same session the classifier was trained on. For each session, leave-two-out-style validation was carried out, where two randomly selected repetitions of each gesture were held out, and the classifiers were trained with the dataset $D_{train}^{(y)}$ composed of the remaining $R - 2$ repetitions of all classes $y \in \mathcal{Y}$. Consequently, the same spatial transformation was applied to the activity images of the held-out repetitions. One transformed held-out repetition of all gestures $y \in \mathcal{Y}$ was used as the test dataset. $D_{test}^{(y)}$. The other transformed held-out repetitions (one per class) were used as the adaptation dataset $D_{adapt}^{(y)}$, where $y \in \mathcal{Y}_{sub}$. The experiment was carried out for the different $\mathcal{Y}_{sub}$ previously defined for the CSL and Capgmyo datasets. In this experiment, a trial was considered one recording session/subject pair a trial, resulting in $N_{sess} \times S$ trials per condition.

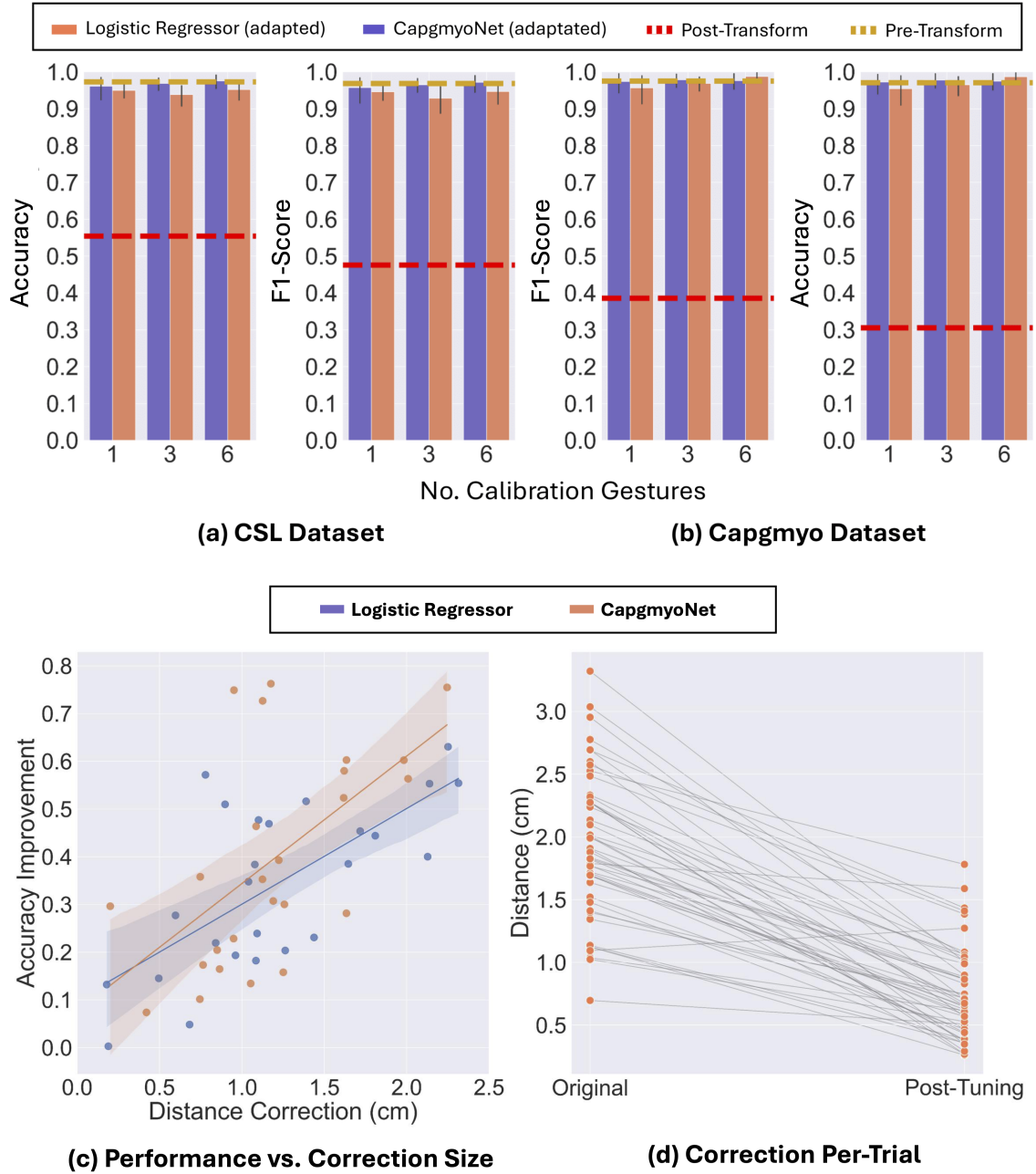


Figure 4.2: A compilation of results from Experiment I, investigating the ability of the Spatial Adaptation Layer (SAL) to mitigate the effects of simulated spatial transformations: **(a,b)** The accuracy and f1-score reported with and without SAL-based domain adaptation (DA) across classifiers, number of calibration gestures and datasets. While the Post-Transform baseline (no adaptation) resulted in low performances, SAL-based DA performances were comparable to intrasession performance (Pre-Transform) across conditions. **(c)** the relationship between accuracy improvement of SAL-based DA with three calibration gestures relative to the baseline. Across classifiers, a significant linear correlation was observed between accuracy improvements and the spatial transformation magnitude. **(d)** The average electrode distance before and after adaptation paired per trial, showing a significant reduction of average electrode displacement.

These spatial transformations were parametrized in the same form as SAL, and act as a controlled use-case where the ground truth spatial transformation is known. For a given recording session

of training data, one repetition of each gesture was randomly held out, and a spatial transformation was applied to this held-out data by randomly sampling from $T_x, T_y \in [-2.5\text{ cm}, 2.5\text{ cm}]$, $\alpha \in [-15.0^\circ, 15.0^\circ]$, $s_x, s_y \in [\frac{1}{1.1}, 1.1]$ and $sh_x, sh_y \in [-0.1, 0.1]$.

The objective was to learn a mapping that indirectly inverts the effects of these spatial transformations (i.e. $A'_\phi \approx A'^{-1}$) and the responsible parameters were directly accessible. Thus, by applying these two transformations consecutively ($A' \cdot A'_\phi$), slicing it to get a 2×3 matrix, and applying to the output grid, the final sampling grid coordinates $[m_i^{final}, n_i^{final}]^T$ were obtained. The more successful the inversion of the originally applied transformation, the closer final coordinates will be target coordinates. Thus, to measure this, the average pixel distance between final and target coordinates were computed, referred to in this study as average electrode distance (d).

$$d(m^t, n^t, m^{final}, n^{final}) = \frac{1}{HW} \sum_{i=1}^{HW} \sqrt{(m_i^t - m_i^{final})^2 + (n_i^t - n_i^{final})^2} \quad (4.17)$$

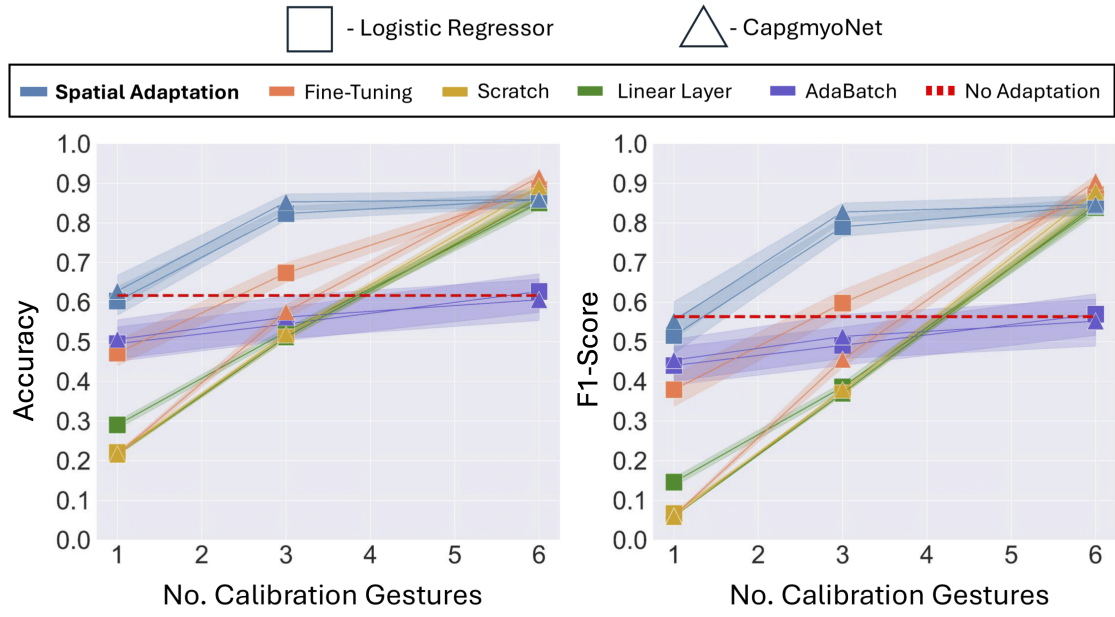
This metric captures the electrode displacement extent post DA, where $d = 0$ when $A'_\phi = A'^{-1}$.

Since performances in Figure 4.2a were not significantly different across classifiers, the average accuracy and f1-score were averaged across trials and classifiers for the Pre-Transform condition, corresponding to classification performance prior to the simulated transformation (equivalent to intrasession performance), and Post-Transform condition, corresponding to classification performance after the simulated spatial transformation without any DA.

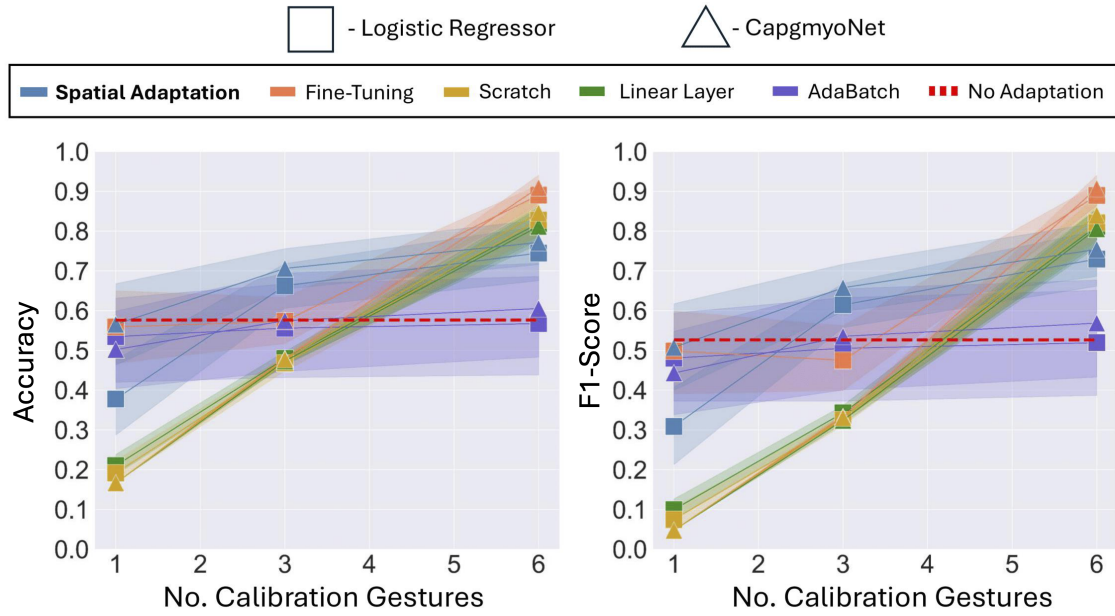
When spatial transformations were applied, classification performance dropped significantly across trials, with performance averaged across classifiers dropping by $\Delta = (-41.9\%, -49.3\%)$, and $\Delta = (-58.9\%, -66.5\%)$ for CSL and Capgmyo datasets respectively. Furthermore, SAL-based DA was considered, performances were consistently high (accuracy and f1-score were $> 90\%$) across classifiers, datasets and number of gestures. However, that since the Capgmyo dataset uses sensors with irregular IED, spatial transformations are applied in a physically unrealistic way, hence were easier to invert than electrode displacement in real data distributions.

For evaluating the interpretability of SAL-based DA, the average electrode distance d was computed across trials. In this analysis, only data from the CSL dataset was considered, since it requires a constant IED for physically meaningful calculations. Furthermore, the Accuracy Improvement was considered, computed as the difference between the accuracy before and after SAL-based DA, reflecting a per-trial performance improvement. As shown in Figure 4.2b, for both classifiers, a significant, moderate to strong positive correlation with the distance correction (d) and the accuracy improvement was observed ($R = 0.584$ and $R = 0.689$ for CapgmyoNet and logistic regressor respectively).

In addition, in Figure 4.2c, shows the paired-per trial difference in (d) for the logistic regressor. Here, two sampling coordinates were compared: post-transform (Original), where ($A_\phi = I$), and post-transform with SAL-based DA, where the optimal ϕ was identified. This effectively measured the effect of SAL-based DA on reducing the average electrode displacement per-trial. For the majority of trials, this metric was greatly reduced, with $> 75\%$ of trials showing post-adaption



(a) CSL Dataset



(b) Capgmyo Dataset

Figure 4.3: Inter-session performances across classifiers, adaptation methods, number of calibration gestures and datasets in Experiment II, with a baseline no adaptation baseline (Post-Transform) averaged across classifiers is displayed in red: **(a)** For the CSL dataset, which uses HD-sEMG arrays with constant interelectrode distance, while Spatial Adaptation Layer (SAL) based domain adaptation (DA) performed comparably when adapting on all available gestures, it significantly outperformed other methods when adapting on fewer gestures. **(b)** For the Capgmyo dataset, which uses HD-sEMG arrays with variable interelectrode distance, while SAL-based DA outperforms other methods when adapting on fewer gestures, its performance was significantly limited in relation to reported performances for CSL.

distances of under 1 cm and the average metric dropping from 1.98 cm to 0.78 cm.

4.4.2 Experiment II: Intersession Performance

To evaluate the ability of **SAL** to compensate for intersession differences, I carried out a **DA** experiment where, for each subject, every valid permutation of two recording sessions was considered a single trial (e.g. Subject i , Session $j \rightarrow$ Session k). Therefore, this experiment contained $S \times N_{sess} \times (N_{sess} - 1)$ trials per condition. A classifier was trained on all available data from one session ($D_{train}^{(y)}, y \in \mathcal{Y}$). For the adaptation set ($D_{train}^{(y)}$), I selected on repetition of each gesture from a pre-defined gesture subset $y \in \mathcal{Y}_{sub}$ from the other recording session. The remaining repetitions ($R - 1$) of the other session for all available gestures (i.e. $y \in \mathcal{Y}_{sub}$) used as the test set $D_{test}^{(y)}$. This experiment was run independently for CSL and Capgmyo datasets and adaptation gesture subset considered ($|\mathcal{Y}_{sub}| \leq |\mathcal{Y}|$).

Beyond **SAL** and zero-shot performance, four other alternatives were considered: fine-tuning, **AdaBatch**, input linear layer and training from scratch. The last of the four options was included as a control. Given that aforementioned non-iid sampling structure in **sEMG**, it is unfeasible to sample individual activity images of each class and must instead sample entire repetitions of a single gesture. Hence, if all classes are sampled $\mathcal{S}$ (i.e. $|\mathcal{Y}_{sub}| = |\mathcal{Y}|$), **DA** methods are expected result in functionally higher performance than training a new classifier from scratch with the same amount of target data.

Furthermore, I constrained **SAL** parameters to $T_x, T_y \in [-5.0\text{px}, 5.0\text{px}]$, $\alpha \in [-15.0^\circ, 15.0^\circ]$, $s_x, s_y \in [\frac{1}{1.1}, 1.1]$ and $sh_x, sh_y \in [-0.1, 0.1]$. While for the CSL dataset, one pixel corresponds to 1 cm (**IED** = 1 cm), for the irregular structure of Capgmyo there is not analogous correspondence between pixels and physical distances. I chose especially large constraints for translations due to unusually large values detected empirically for the CSL dataset. Performance metrics for each different number of available adaptation gestures, $|\mathcal{Y}_{sub}|$, and dataset are shown in Figure 4.3 for each adaptation method and classifier. As a pre-adaptation baseline measure, I aggregate performance metrics across classifiers to get a single baseline for each dataset/metric.

For the CSL dataset, which uses a **HD-sEMG** grid with constant **IED**, when adapting on all available gestures ($\mathcal{Y}_{sub} = \mathcal{Y}$), while the fine-tuned **CNN** exhibits the highest average performance, it does not lead to large improvements over **SAL** or training from scratch ($\Delta < 2.5\%$ relative to both methods with either classifier). Additionally, the **SAL** adapted logistic regressor performs comparably to the fine-tuned **CNN** ($\Delta = (2.9\%, 3.0\%)$). The only method that performed significantly worse relative to others is **AdaBatch** ($\Delta = (27.4\%, 30.4\%)$ relative to fine-tuned **CNN**). On the other hand, for the Capgmyo dataset, **SAL** exhibited lower performance than all other adaptation methods ($\Delta = (14.3\%, 16.4\%), \Delta = (12.1\%, 14\%), \Delta = (4.9\%, 6.8\%)$ relative to fine-tuning, scratch training and linear layer respectively) excluding **AdaBatch**. When I considered $|\mathcal{Y}_{sub}| = 1$, all adaptation methods exhibit worse performance than the baseline across datasets.

However, when I considered $|\mathcal{Y}_{sub}| = 3$, while most other adaptation methods remain below the baseline, **SAL**-based **DA** resulted in large performance improvements for both datasets relative to the baseline ($\Delta = (22.0\%, 24.3\%)$ and $\Delta = (14.5\%, 15.9\%)$ with CapgmyoNet for CSL and

Table 4.1: Performance by number of adaptation gestures, classifier, median filtering, and Baseline normalization (BN) method.

$ \mathcal{Y}_{\text{sub}} $	Classifier	Median	Accuracy			F1-score		
			BN-MS	BN-RMS	No BN	BN-MS	BN-RMS	No BN
1	CapgmyoNet	Filt.	0.671	0.672	0.672	0.604	0.607	0.605
		No Filt.	0.661	0.671	0.515	0.593	0.609	0.412
	Logistic Regressor	Filt.	0.617	0.635	0.603	0.539	0.562	0.519
		No Filt.	0.644	0.645	0.517	0.571	0.573	0.408
3	CapgmyoNet	Filt.	0.866	0.861	0.847	0.845	0.837	0.819
		No Filt.	0.855	0.843	0.822	0.834	0.818	0.789
	Logistic Regressor	Filt.	0.843	0.837	0.821	0.819	0.810	0.785
		No Filt.	0.821	0.820	0.708	0.790	0.792	0.632
6	CapgmyoNet	Filt.	0.898	0.889	0.873	0.889	0.877	0.860
		No Filt.	0.878	0.864	0.856	0.867	0.850	0.838
	Logistic Regressor	Filt.	0.874	0.874	0.867	0.863	0.846	0.854
		No Filt.	0.870	0.863	0.849	0.858	0.846	0.831

Capgmyo datasets respectively). The only method that outperforms the baseline is fine-tuning the logistic regressor.

4.4.3 Experiment III: Ablation Study

In this [ablation study](#), the aim was to determine the effect of different baseline normalization methods on [SAL](#)-based [DA](#) performance, as well as their interactions with spatial median filtering, which is used to handle isolated corrupted channels in an activity image. Thus, I carried out the same protocol for as for Experiment II, but considering each baseline normalization method (or none), with or without spatial median filtering. For conciseness, I only considered results for [CSL](#), as the [HD-sEMG](#) arrays used have constant [IED](#). From results in Table 4.1, while **BN-MS** results in higher performances for the highest number of conditions, performances between **BN-MS** and **BN-RMS** were comparable, and do not vary by more than 0.5% from the **No BN** condition when using the spatial median filter.

4.4.4 Experiment IV: Effect of Hybrid Optimization Strategy

To evaluate the effect of the gradient-based optimization after the initial brute-force search, I evaluated [SAL](#)-based [DA](#) in the same experimental setting as in Experiment II across different number of gestures and classifiers, but considering the hybrid optimization ([LHS](#) with 500 epochs of gradient-based optimization) versus only [LHS](#). As shown in Table 4.2, the addition of the gradient-based optimization in the Hybrid strategy led to $\Delta < 0.5\%$.

4.4.5 Experiment V: Interpretability Analysis

Using the results of Experiment II, magnitude of each learned [SAL](#) parameter was inspected across [DA](#) trials. Across subjects and sessions, I computed the Pearson correlation coefficient (R_p) between the accuracy improvement ($Accuracy_{\text{post}} - Accuracy_{\text{pre}}$) and the magnitude of each parame-

Table 4.2: Comparison for different optimization strategies and classifiers across adaptation set sizes

$ \mathcal{Y}_{\text{sub}} $	Classifier	Accuracy		F1-score	
		Hybrid	LHS	Hybrid	LHS
1	CapgmyoNet	0.666	0.680	0.604	0.614
	Logistic Regressor	0.625	0.653	0.552	0.584
3	CapgmyoNet	0.864	0.867	0.842	0.846
	Logistic Regressor	0.841	0.839	0.815	0.813
6	CapgmyoNet	0.893	0.887	0.882	0.874
	Logistic Regressor	0.881	0.877	0.871	0.865

ter. In this experiment, I only considered data for $|\mathcal{Y}_{\text{sub}}| = 3$ and the CSL dataset, where a constant **IED** ensured parameters were physically meaningful.

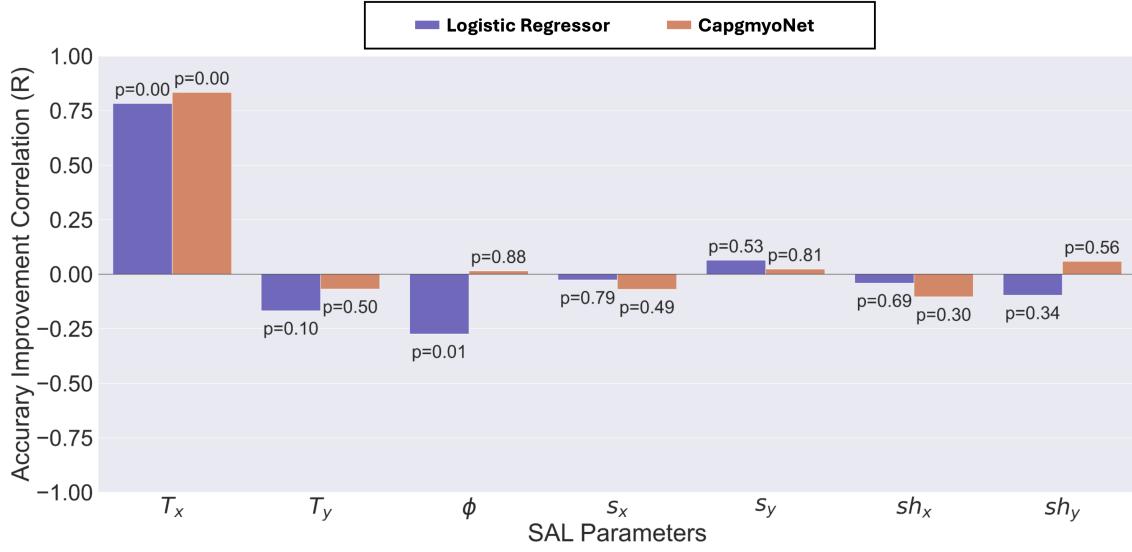
As shown in Figure 4.4a for both classifiers, For most parameters, low R_p values ($R_p < 0.3$) and non-statistically significant p-values ($p > 0.01$) were observed. On the other hand, for the T_x parameter, which reflects transversal displacement, high, statistically significant correlations ($R_p = 76$ and $R_p = 0.83$ for the logistic regressor and CapgmyoNet respectively, $p < 0.01$) were observed. This suggests that the only meaningful parameter for performance improvements in the context of **HD-sEMG** activity image gesture classification is T_x . Additionally, in Figure 4.4b, detection of T_x of up to 4.5 cm displacements was observed, in this case resulting in accuracy improvements of over 60%.

Furthermore, while mostly statistically insignificant, quite a few parameters exhibit negative R_p values. Therefore, I considered a final interpretability experiment, where I use the same experimental protocol for Experiment II, and compare conditions T_x , where T_x is the only learnable parameter of **SAL**, versus **SAL** with all learnable parameters. Given results from Experiment IV, since for the seven parameters **LHS** only seems to be sufficient, it follows that **LHS** only should also be sufficient for **SAL** with a single parameter. Therefore, the T_x only condition does not involve gradient-based optimization.

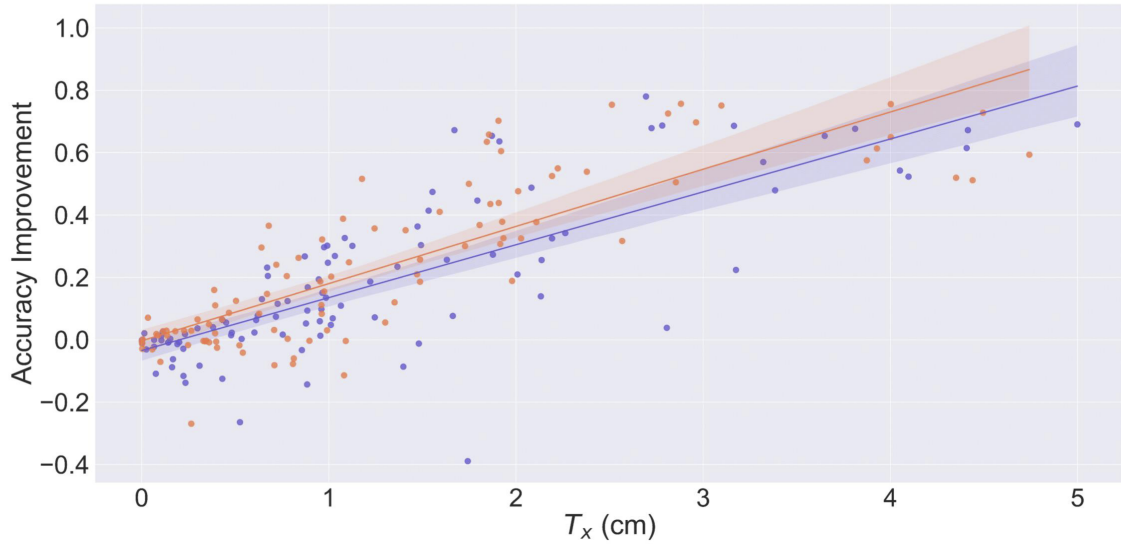
As shown in Table 4.3, considering T_x with only **LHS** optimization results in large performance improvements for $|\mathcal{Y}_{\text{sub}}| = 1$ ($\Delta = (14.1\%, 17.1\%)$), where an accuracy of 81.3% was obtained, small improvements for $|\mathcal{Y}_{\text{sub}}| = 3$ ($\Delta = (1.3\%, 1.8\%)$), and only slightly worse, but comparable performance when all gestures are available for adaptation ($\Delta = (-1.3\%, -1.7\%)$).

4.5 Discussion

The results of this study demonstrated that **SAL**-based **DA** is versatile, effectively improved performance even when provided with a subset of gestures in the target domain, and resulted in interpretable results, where learned parameters corresponded to trends observed in previous works on **sEMG** electrode displacement. Overall, **SAL**-based **DA** is the method that most achieves the calibration method desiderata previously introduced in this study.



(a) Parameter Relevance



(b) Transversal Distance

Figure 4.4: Statistical analysis of the relationship between magnitude of the learned Spatial Adaptation Layer (SAL) parameters across recording sessions and accuracies given the outcomes of Experiment II for the CSL dataset: **(a)** For both classifiers, the only parameter that resulted in a significant positive correlation with accuracy improvements was T_x , representing transversal displacements. **(b)** Significant positive linear correlations are observed between T_x magnitude and accuracy improvement across classifiers, further highlighting results reported in Figure 4.2.

Gesture/Repetition Efficiency: Given observations based on results from Experiment II, when data is available from all classes, even re-training a classifier from scratch with such data is able to achieve meaningful intersession performances (accuracy was $\geq 85\%$) for the CSL dataset, further reflecting that the real benefit of novel calibration methods must be gesture/repetition efficiency. Furthermore, for the CSL dataset, when only three out of six total classes were available for adaptation, almost all adaptation methods and classifiers performed worse than the baseline, reflecting

Table 4.3: T_x only vs all parameters across classifiers and $|\mathcal{Y}_{\text{sub}}|$

$ \mathcal{Y}_{\text{sub}} $	Classifier	Accuracy		F1-score	
		T_x	All	T_x	All
1	CapgmyoNet	0.822	0.666	0.788	0.604
	Logistic Regressor	0.737	0.625	0.692	0.552
3	CapgmyoNet	0.872	0.864	0.851	0.842
	Logistic Regressor	0.851	0.841	0.833	0.815
6	CapgmyoNet	0.880	0.893	0.865	0.882
	Logistic Regressor	0.871	0.881	0.858	0.871

class-imbalance overfitting, except for **SAL**-based **DA**. Lastly, after noticing the negative effect of specific **SAL** parameters on performance in Experiment V, an ablation experiment with only a single **SAL** parameter was conducted. This resulted in functionally meaningful performances for the CSL dataset (accuracy was $\geq 85\%$ for CapgmyoNet and logistic regressor respectively in Experiment V) when adapting on three out of six gestures, and even a nearly functional accuracy level of 82.2% when using CapgmyoNet and adapting on one out of six total classes. Therefore, even a single repetition of a single gesture may be enough to calibrate a subject-specific model to a new recording session when using **SAL**-based **DA**.

Interpretability: From Experiment I, **SAL**'s ability to ensure high performance on data with simulated spatial transformations was consistently high across datasets, classifiers and number of calibration gestures available (accuracy $> 90\%$) suggests that it can indeed mitigate the effects of electrode displacement despite optimizing parameters only via a classification loss. Furthermore, the average electrode distance was significantly reduced by **SAL**-based **DA**, suggesting that **SAL** is tackling the true underlying electrode displacement.

Additionally, the correlation analysis in Experiment V suggested the only meaningful parameter for performance improvements in the context of **HD-sEMG** activity image gesture classification is T_x , in alignment with previous studies [157]. This is further supported by the following ablation experiment, where **SAL** with only T_x resulted in comparable results for most conditions to **SAL** with all parameters, and significantly higher results for one out of six gestures available for adaptation. Since the **HD-sEMG** grids used were not flexible, scalings or shearing effects were hypothesized to not contribute to adaptation. In addition, large rotations are unlikely when placing grids along the circumference of the forearm. Likewise, longitudinal translations follow the length of forearm muscles, carrying largely similar information when considering the space of activity (**RMS**) images. Hence, the correlation analysis reflected expectations of the physical transformations the **HD-sEMG** arrays were likely to have experienced across recording sessions. Thus, **SAL**-based **DA** is only dependent on seven parameters that learn physically meaningful information despite only depending on classification loss.

Low Computational/Memory Overhead: While dependent on **LHS** brute-force optimization, since activity images were aggregated per class, **SAL** optimization still remains highly tractable. Furthermore, by being classifier independent, it enables the use of deep learning models, as well

as smaller, more interpretable models, such as a logistic regressor. Not only does the combination of logistic regressor and SAL only have a few thousand parameters (given the datasets explored in this study), but resulted in an accuracy $\geq 85\%$ for even just three out of six available gestures for adaptation, comparable to fine-tuning a CNN given the protocols in Experiment II, III, and V. Hence, SAL optimization is tractable, and using SAL with simple, non-deep classifiers enable gesture classification pipelines with low computational and memory overhead.

Versatility: SAL-based DA exhibited a much larger degree of versatility than traditional electrode displacement calibration methods, as all you need is supervision via classification loss. However, despite being more gesture efficient than other DA methods considered, SAL did not reach performances that constitute a useful sEMG gesture classification system when applied to an sEMG array with variable IED. This reflects how, given its current assumptions, SAL works optimally with an acquisition set-up with constant IED, which may be limiting regarding all possible HD-sEMG acquisition set-ups.

Preprocessing For Non-Spatial Variability: Beyond evaluating SAL against the core desiderata, several preprocessing and optimization components were also considered to determine the optimal overall pipeline. Ablation studies revealed that baseline normalization methods (BN-MS and BN-RMS) contributed minimally to performance, particularly when spatial median filtering was employed. Similarly, while SAL’s sub-differentiability enables gradient-based optimization, experiments showed that thorough brute-force search alone achieved comparable results without additional computational overhead.

These pipeline optimizations reveal that the effectiveness of SAL stems not just from its spatial transformation approach, but from its compatibility with simple, efficient processing chains. The minimal impact of baseline normalization methods and the sufficiency of brute-force optimization suggest learning transversal displacement with SAL is robust enough to provide high performance (accuracy was $\geq 85\%$) without extensive preprocessing or complex optimization strategies. This simplicity, combined with meaningful performance using limited calibration data, addresses three of the four desiderata effectively. However, the constant IED requirement represents a fundamental constraint that limits SAL’s applicability compared to model-based adaptation methods that can handle arbitrary electrode configurations.

4.6 Conclusion

In this study I introduce SAL, a novel input-based DA approach for addressing electrode displacement in HD-sEMG gesture classification. SAL is designed to meet four desiderata: gesture efficiency, low computational overhead, interpretability, and method versatility.

SAL successfully achieves the first three desiderata. It demonstrated strong adaptation when only a small subset of gestures is available, yielding 22.0% and 14.5% accuracy improvements over alternative methods on the CSL and Capgmyo datasets when using just three out of six gestures. The interpretability analysis conducted suggests that transversal displacement (T_x) is the only parameter strongly correlated ($R_p = 0.76 - 0.83, p < 0.01$) with performance improvements, consistent

with physiological expectations [157]. This insight enables the use of a streamlined SAL configuration using brute-force search with T_x as the sole learnable parameter, achieving nearly functional performances single-gesture calibration (82.2% accuracy) while minimizing computational burden. This streamlined version even supports effective performance (accuracies of $> 85\%$) when adapting on three out of six gestures with a lightweight logistic regression classifier.

In terms of versatility, SAL occupies a middle ground. It is less flexible than broader DA methods, which can address diverse forms of intersession variability, yet it is more versatile than prior direct electrode displacement calibration techniques, providing a physically interpretable and computationally efficient pathway to improved generalization. However, its reliance on constant IED currently limits applicability to regular electrode arrays, with weaker performance observed on irregular arrays. Therefore, future work should extend SAL to irregular configurations.

Crucially, SAL-based DA achieves all the three classification success criteria defined in Chapter 1: an acceptable response time (due to the chosen window size), as well as classification accuracies of $\geq 85\%$ with an adaptation set that includes only three gestures. By mitigating electrode displacement, even a logistic regressor classifier met the success criteria, despite its linear constraints. This suggests that SAL can be a useful integration for gesture classification myoelectric control systems to provide robustness across real-world conditions.

In summary, SAL-based DA is a principled, interpretable method for intersession DA that effectively reduces the challenge of intersession generalization to that of intrasession generalization. SAL can be integrated into regular grid HD-sEMG systems, demonstrating that physically meaningful parameters can be learned directly through classification loss while offering a practical balance between interpretability, efficiency, and versatility.

Chapter 5

Tracking Motor Units Across Sessions: A Filter Adaptation Problem

5.1 Introduction

5.1.1 Motor Unit Decomposition

During movement, each motor neuron and the fibres it innervates, known as a **motor unit (MU)**, activate to generate force. The sensing of electric potential differences due to the superposition of these activations is known as electromyography (EMG). When sensed from the skin surface or minimally invasively with concentric needles, the approach is referred to as **surface electromyography (sEMG)** or **intramuscular electromyography (iEMG)** respectively. This sensing modality has facilitated advancements in understanding of motor control and neuropathy, as well as in naturalistic control of prostheses and exoskeletons [108].

One popular processing method of an **high-density surface electromyography (HD-sEMG)** electrode array is **MU decomposition** [48], where the firing times of a subset of **MUs** are estimated from **HD-sEMG**. One of the most popular means of decomposition is to model **HD-sEMG** as a linear convolutive mixture of sparse (spiking) sources [70], where **blind source separation (BSS)** techniques such as fastICA can be used to separate sources given their sparsity [19]. Once the separation procedure is complete, the learned separation filters can be used in real-time for detecting **MU** discharge times. The simultaneous measurement of a large number of **MUs** enables the direct study of **MU** physiology, as well population characteristics and firing patterns [48]. Furthermore, efferent motor commands at the output of the spinal cord can be estimated through the cumulative spike train, referred to as the effective neural drive [52]. This information can be used for investigating corticospinal transmission and integrity [77], or even directly as control signals for assistive devices.

5.1.2 Motor Unit Tracking & Limitations

While a popular tool in neurophysiology and myoelectric control, **MU** decomposition has drawbacks in real-world settings. The learned separation filters counter the effect of volume conduc-

tion: the spatiotemporal filtering carried out during the propagation of MU activity from the source to the electrode location. Thus, changes in electrode placement, noise distribution and contraction level are some of numerous factors that affect the spatiotemporal filtering, often rendering original separation vectors redundant.

Therefore, MU tracking, which aims to find which physical MU from one recording condition corresponds to which MU in another recording condition, is used to match MUs and analyse their firing and recruitment properties across these conditions. While some studies focus on tracking MUs across contraction levels [7], the most difficult tracking challenge remains the tracking of MUs of the same subject and muscle across recording sessions.

The current state-of-the-art approach for MU tracking across recording sessions involves applying MU decomposition independently to two different recording sessions, computing the spike-triggered average (STA) estimate of the motor unit action potential (MUAP) templates, and computing the temporal cross-correlation of these between MUs from the first and second session [100]. However, their approach exhibits four limitations:

1. Clear markings were drawn on the skin of the subjects at the original grid location, preventing electrode displacement. This implies that without physical markings under more realistic conditions this approach is limited.
2. Due to large amounts of spatial correlation, normalized cross-correlation can be large even between different MUs, as shown by the relatively large number of MUs that had multiple candidates reaching supra-threshold cross-correlations ($\approx 15\%$ of trials) [100].
3. Since MU decomposition is performed independently for each recording session, it is unlikely that the exact same subset of MUs is extracted from the MU pool from both sessions, significantly lowering the number of MUs that can be successfully tracked.
4. MU decomposition is notorious for the amount of manual editing required after using different automatic BSS algorithms. By decomposing independently, manual editing is performed for every recording session without leveraging previously edited decomposition results.

5.1.3 Motor Unit Tracking As A Domain Adaptation Problem

MU tracking across recording sessions is fundamentally a problem of non-stationarity when using BSS methods. As with gesture classification in Chapter 4, this problem can be framed as virtual concept drift: the model parameters θ_S optimized on the first recording session sampled from the source domain ($\mathcal{D}_S = P_S(X, Z)$) are sub-optimal when applied to the second recording session sampled from the target domain ($\mathcal{D}_T = P_T(X, Z)$).

In this context, instead of matching MUs independently decomposed through waveform cross-correlations, I propose the use of the input-based domain adaptation (DA) framework from Chapter 4 to mitigate the effects of non-stationarities in motor unit tracking. Since authors in [100] were able to track a substantial proportion of MUs across recording sessions through electrode displacement prevention by drawing markings on the skin of their subjects, I assume the primary cause of non-stationary is electrode displacement. To mitigate this, the Spatial Adaptation Layer

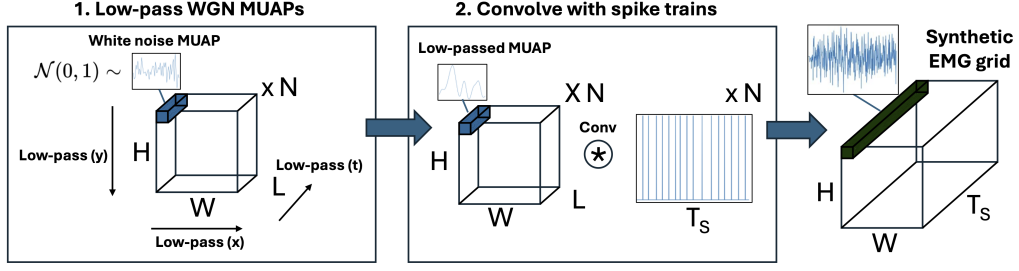


Figure 5.1: Schematic of the steps to simulate sEMG and apply spatial transformation. **(1)** white Gaussian noise based motor unit action potentials (MUAPs) were sampled from $H \times W \times L$ from $\mathcal{N}(0, 1)$, and lowpassed along every dimension with respective cutoff frequencies. **(2)** Independent spike trains were generated for each of N motor units (MUs), and for every channel, summed the convolution of all spike trains with corresponding lowpassed MUAPs. This resulted in a synthetically generated, regular HD-sEMG grid. In practice, the system was initially oversampled with a scaling factor κ , only downsampling after the appropriate transformations were applied.

(SAL) Chapter 4 is employed to learn the optimal translations and rotations of input HD-sEMG signals that ensure the original separation vectors are BSS-optimal in a new electrode placement. This addresses all four limitations introduced:

1. The pipeline is used to mitigate the effects of electrode displacement, hence can be used in more realistic intersession conditions.
2. As opposed to comparing MUAP waveforms, this method finds the unique, constrained BSS-optimal estimate of the source, effectively filtering out spurious, non-optimal matches.
3. The use of all separation vectors from the first recording session reduces the dependence of the tracking on which subset of MUs was decomposed during the second recording session.
4. Using the same manually edited and visually validated MUs, the proposed method reduces the necessity of manual cleaning and visual inspection for subsequent recording sessions.

5.2 Methods

5.2.1 Data

Phemenological EMG Simulations

Simulating HD-sEMG Grids: Since decomposition is a BSS problem, synthetic HD-sEMG with realistic properties were generated such that a convolutive mixture was obtained with the ground truth MU sources known. In this simulation set-up, a high-density grid of 26×10 electrodes was considered with interelectrode distance of 0.4 cm ($f_s^{(x,y)} = 250 \text{ m}^{-1}$), inspired by a 256 channel grid (GR04MM2610, OTBioelettronica, Torino, Italy). A temporal sampling rate of $f_s^t = 2000 \text{ Hz}$ was used, along with a MUAP length of $L = 50$ and acquisition duration of 10s.

Synthetic MUAPs: By treating the HD-sEMG grid as a 3D signal, composed of one temporal (t) and two spatial (x, y) dimensions, MUAPs were modelled as white Gaussian noise (WGN) band-limited along each of three dimensions, which was practically obtained by applying a Butterworth

digital lowpass filter (second order) along each of the three dimensions of **MUAPs**. While this **MUAP** model does not capture the full physiological complexity of real **HD-sEMG** signals, band-limiting in both spatial and temporal dimensions ensures that the synthetic signals exhibit the key properties relevant to this evaluation: smooth spatial correlation structure across the grid and realistic temporal bandwidth. Spike trains were generated independently for each **MU** as a stochastic renewal process with a Gaussian distribution with average time interval (T_{mean}) of 60 samples and interspike variability (ISV) of 0.2 [107]. After simulating the synthetic **MUAPs**, to obtain the complete, synthetic **HD-sEMG** grid, the **MUAPs** of each **MU** over all channels were convolved with respective spike trains and summed the resultant signals.

Simulation Robustness: To limit the effects of truncation on the IIR filters used for lowpassing (small number of samples), and to limit the interpolation error from bilinear interpolation, the 3D signal was initially oversampled across spatial and temporal dimensions by a factor of κ , increasing the initial effective sampling rates to κf_s^t and $\kappa f_s^{(x,y)}$ respectively. After band-limiting and transforming, the signal was downsampled along the temporal and spatial dimensions by a factor of κ . In this study, $\kappa = 14$ was used.

Simulation Pipeline: Thus, as shown in Figure 5.1, **MUAPs** were initially oversampled from **WGN** and lowpassing along temporal and spatial dimensions. Band-limited **MUAPs** were then convolved with corresponding spike trains to obtain the synthetic **HD-sEMG** grid. Downsampling was only applied after the appropriate transformations. Temporally, a lowpass cutoff of $f_c^t = 500\text{Hz}$ was used, a commonly assumed upper cutoff frequency for **HD-sEMG** bandwidth [72, 19, 9]. For spatial properties, the conduction velocity of $v = 4\text{ms}^{-1}$ was initially considered for muscle fibres parallel to the skin surface based on a biophysical model of the limb [46]. This would imply a spatial cutoff frequency along the longitudinal direction of a limb muscle of $f_c^{(x,y)} = 500/4 = 125\text{m}^{-1}$ [46]. However, this estimate was dependent on each subject's anatomy and fibre orientation. Therefore, in this study, $f_c^{(x,y)}$ was treated as a simulation parameter (referred to as f_c for simplicity) and considered within the range $f_c \in [62.5, 182.5]\text{m}^{-1}$. Note that conditions of effective spatial aliasing were considered due to the initial oversampling.

Experimental Data ¹

Participants: Three healthy, adult male participants with no known history of neurological or musculoskeletal disorders were recruited for this study. All participants provided written informed consent prior to their inclusion. The study was conducted in accordance with the principles outlined in the Declaration of Helsinki and received approval from the Imperial College London Ethics Committee (reference number 18IC4685).

Experimental Design: Participants were seated in a chair with their knee fully extended and their leg strapped to an ankle dynamometer. The foot was positioned on a pedal inclined at 30° in the plantar-flexion direction (where 0° corresponds to the foot being perpendicular to the shank). Initially, participants performed a single **maximum voluntary contraction (MVC)** of the ankle with a dorsiflexion, serving as a reference for the following tasks. Visual feedback was provided

¹The acquisition protocol design, data collection and preprocessing were carried out by Emanuele Abbagnano.

throughout the experiment, with a target displayed on a screen representing the target contraction level and a real-time trace showing the force exerted by the participant. Each participant performed a total of two isometric dorsiflexion contractions per tested %MVC with the **Tibialis Anterior (TA)** muscle. These included a linear ramp phase where force increased from 0% to the target %MVC level at a rate of 5% MVC per second, followed by a plateau phase at the target level, and finally a phase back from the plateau to 0% MVC at the same rate (trapezoidal force profile). Two target %MVC levels were used: 25% (20 s plateau) and 50% (15 s plateau). The duration of these plateaus was chosen based on previous studies showing successful decomposition for segments longer than 5 s at sub-maximal contraction levels [33]. There was a 2 min rest period in between each contraction to reduce fatigue. Subjects performed the experiment on three separate days, with the second recording session being 7 days after the first, and the third recording session being 21 days after the first.

Data Acquisition: Four surface grids of 64 electrodes (13×5) (GR04MM2610, OT Bioelettronica, Torino, Italy) were placed contiguously over the muscle belly of the **TA** along the fiber orientations on the self-reported dominant leg. This formed a larger virtual grid (26×10). The **interelectrode distance (IED)** of each grid was 0.4 cm. While the same placement strategy was used for every recording session, no physical markings were left on the skin, reflecting realistic electrode displacement across placements. EMG signals were acquired with a monopolar derivation, amplified using the Quattrocento Amplifier system (OT Bioelettronica), sampled at 2048 Hz. With the same amplifier system, force signals were acquired via a transducer (TF-022, CCt Transducer s.a.s.) mounted on the pedal of the ankle dynamometer and digitalized at 2048 Hz.

Data Preprocessing: Signals were processed with a digital bandpass filter (20-500 Hz, second-order Butterworth) and power-line interference was reduced using a digital bandstop filter (45-55 Hz, second-order Butterworth). Outlier channels were identified per recording session with visual inspection. Once identified, outlier channels were iteratively replaced with the average of all non-outlier neighboring channels, starting with outlier channels with the highest number of non-outlier channels.

5.2.2 Motor Unit Decomposition ²

Convolutional Blind Source Separation

Multichannel EMG signals recorded during the experimental tasks were decomposed offline by Emanuele Abbagnano using the convolutional **BSS** algorithm from [116] (detailed in Chapter 2.4), resulting in separation vector estimates and real-valued source estimates for a subset of active **MUs**.

Source Estimate Post-Processing

Given a separation matrix, $\mathbf{B} \in \mathbb{R}^{N \times MR}$ sources are estimated as a real-valued matrix $\mathbf{T} \in \mathbb{R}^{N \times T_s}$. Since the aim of decomposition is to obtain discharge times, this must be processed to obtain the sources as a series of binary spike trains, $\mathbf{T}_b \in \{0, 1\}^{N \times T_s}$. To achieve this, source vectors were

²The motor unit decomposition procedure and manual spike editing were performed by Emanuele Abbagnano.

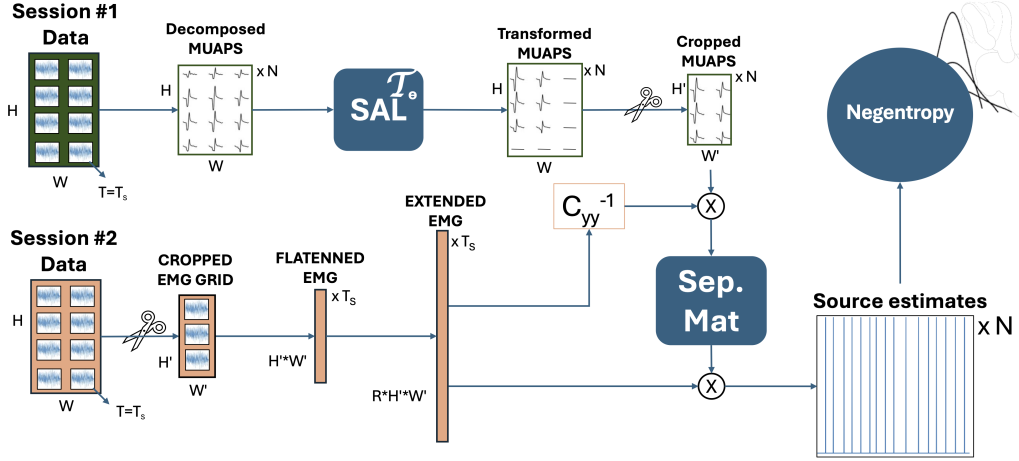


Figure 5.2: Pipeline of sub-differentiable spatial adaptation for motor unit (MU) decomposition. Firstly, the linear minimum mean square error (LMMSE) optimal separation matrix was split into two components: the inverse covariance matrix of extended observations, and spike-triggered average (STA) templates for all decomposed MUs. **First recording session:** the STA motor unit action potential (MUAP) templates were estimated for the first recording session. A central sub-region of the transformed MUAPs was cropped, such that no post-transformation out-of-bound electrodes were included. **Second recording session:** the second recording session was cropped in the same manner as the MUAPs from the first session. Subsequently, observations were extended and inverse covariance matrix of was estimated. **Pipeline Output:** The matrix product of the cropped STA templates (session 1) and inverse covariance matrix of extended observations (session 2) was applied as a separation matrix to extended observations from the second recording session, resulting in source estimates. From these, the average source negentropy ($J(\mathbf{x})$), a common blind source separation (BSS) learning objective, was computed. Thus, this pipeline provided a sub-differentiable learning objective that could be used to determine the spatial parameters that were BSS-optimal.

squared, a peak detection algorithm was applied, and the K-means++ algorithm [3] was applied to each source estimate independently with two clusters, where the cluster with the higher centroid value was chosen as the spikes for the given source.

Manual Spike Editing

Spike trains were then visually inspected and manually edited by an expert operator (Emanuele Abbagnano), following established guidelines [33]. This implied the addition of missed discharges, and the removal of false positive predicted discharges, as to ensure enough firing regularity, as expected in realistic physiological conditions. To discard erroneously detected MUs, the silhouette (SIL) value was used to quantify the decomposition confidence by comparing the distance within and between spike and non-spike clusters [116].

5.2.3 Decomposition Refinement

From the predicted discharge times, linear minimum mean square error (LMMSE) optimal separation vectors can be estimated (more detailed in Chapter 2.4.3). Thus, the estimated discharge times were refined in the same manner as [116], where new separation vector estimates result in new discharge time estimates, which enable the estimation of yet another set of separation vectors.

This refinement procedure is known to improve firing regularity and decomposition performance. Thus, initial separation vectors for the adaptation pipeline were obtained from discharge times estimates after refinement.

5.2.4 Spatial Decomposition Adaptation

Spatial Adaptation Layer

In Chapter 4, I formulated **SAL** as a transformation that directly mitigates a uniform spatial transformation to a regular grid. Under a **DA** formulation, given a small amount of data from a second recording session sampled from $\mathcal{D}_T$, the aim is to a model with parameters optimized on the first recording session sampled from $\mathcal{D}_S$.

While this was validated on a gesture classification task, this framework is valid for any differentiable loss function. Thus, in this study, I implement **SAL** into a **Spatial Decomposition Adaptation (SDA)** pipeline, with the aim of finding the uniform spatial transformation that minimizes the **BSS**-inspired learning objective.

In this study, for all experiments, an affine coordinate transformation A_ϕ parametrized by parameters $\phi = [T_x, T_y, \alpha]^T$ was considered, corresponding to transversal translations, longitudinal translations and rotations respectively. Additionally, transformation parameter bounds were defined as $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$, and $\alpha \in [-15^\circ, 15^\circ]$.

Mitigating Non-Spatial Intersession Variability

While electrode displacement is assumed to be the largest confounding factor between recording sessions, variability due to electrode impedance and noise distribution can result in substantial deviations on the inverse covariance matrix of extended observations relative to the first recording session.

For the second recording session, the **LMMSE** optimal separation matrix (see Section 2.4.3 for more detailed information) can be estimated as:

$$\mathbf{B}_2^* = \mathbf{S}_{\tilde{\mathbf{y}}_2} \mathbf{C}_{\tilde{\mathbf{y}}_2}^{-1} \quad (5.1)$$

The physical changes between recording sessions were modelled through a function h , where $\tilde{\mathbf{y}}_2(k) = h(\tilde{\mathbf{y}}_1(k))$. Under the assumption that h was a linear operator, it can be inferred that:

$$\mathbb{E}[h(\tilde{\mathbf{y}}[k])] = h(\mathbb{E}[\tilde{\mathbf{y}}[k]])$$

This enabled the modelling of the **STA** matrix from the second recording session given the **STA** matrix from the first recording session as:

$$\hat{\mathbf{S}}_{\tilde{\mathbf{y}}_2} = h(\mathbf{S}_{\tilde{\mathbf{y}}_1}) \quad (5.2)$$

Under the assumption that the effect of h is primarily electrode displacement, its effects were mitigated using [SAL](#) to learn a function g_ϕ . In this context, separation matrix was represented as a function parametrized by spatial parameters ϕ :

$$\hat{\mathbf{B}}_2(\phi) = g_\phi(\mathbf{S}_{\tilde{\mathbf{y}}_1})^T \mathbf{C}_{\tilde{\mathbf{y}}_2 \tilde{\mathbf{y}}_2}^{-1} \quad (5.3)$$

The above formulation is [LMMSE](#) optimal when it results in the source estimate with the lowest mean squared error. Since the sources are sparse spike trains, the sources are highly non-Gaussian. For this reason, inspired by [BSS](#)-based [MU](#) decomposition, the optimal ϕ was defined as:

$$\phi^* = \arg \min_{\phi} \mathcal{L}(\mathbf{T}(\phi)) \quad (5.4)$$

where the source estimates are given by:

$$\hat{\mathbf{T}}(\phi) = \hat{\mathbf{B}}_2(\phi) \mathbf{Y}_2 = g_\phi(\mathbf{S}_{\tilde{\mathbf{y}}_1}^T) \mathbf{C}_{\tilde{\mathbf{y}}_2 \tilde{\mathbf{y}}_2}^{-1} \mathbf{Y}_2 \quad (5.5)$$

where $\mathcal{L}(\cdot)$ is a differentiable measure of the non-Gaussianity of the source estimates.

Handling Out-Of-Bounds Channels For Spatial Adaptation

Since [SAL](#) applies a virtual spatial transformation by transforming grid coordinates and resampling, when transformed grid coordinates are out of bounds, the resampling is enabled with zero-padding. This not only creates differences between [STA](#) templates and the data, but also creates a mismatch between the covariance of the transformed data and the ground-truth electrode displaced recording session. When the transformation is large enough, these deviations can result in substantial changes on the inverse covariance matrix of extended observations ($\mathbf{C}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}^{-1}$). Recomputing the inverse covariance matrix of extended observations on the fly would be the ideal solution, but the computational complexity of the inversion of large matrices ($\mathcal{O}((MR)^3)$) makes this approach unfeasible.

To mitigate the effect of out-bound pixels, I defined maximum allowable transformations and, based on these, computed the conservative cropping that preserves a central subset of channels, such that the largest allowable transformations such that the transformed coordinates still remained within-bounds of the original, complete grid. In the following notation, the vertical axis of the grid is assumed to be aligned longitudinally with the muscle, whereas the horizontal axis is aligned transversally with the muscle. This was carried out as follows:

1. **Determine Maximal Boundary Constraints:** Construct a largest spatial transformation matrix ($\mathbf{A}(\phi_{max})$) to obtain transformed coordinates $\mathbf{x}' = \{(x'_i, y'_i)\}_{i=1}^{H \times W}$.
2. **Identify Minimal Safe Crops:** For the coordinates $\mathbf{x}'$, perform a brute-force search to identify the minimal integer crops ($c_{left}, c_{bottom}, c_{right}, c_{top}$) on the original grid such that:

$$0 \leq x'_i < W - 1, 0 \leq y'_i < H - 1, \forall i$$

3. **Enforce Conservatism:** Since $-\phi_{max}$ is also a valid transformation, the largest vertical crop was determined ($c_y = \max(c_{top}, c_{bottom})$) and applied it to the top and bottom sides. Likewise, the largest horizontal crop ($c_x = \max(c_{left}, c_{right})$) was determined and applied it to the left and right sides.

For the parameter bounds chosen for the experiments, the determined vertical and horizontal crops were $c_y = 5$ and $c_x = 3$. Given the decrease in channels processed by the [SDA](#) pipeline, this cropping approach has the additional benefit of significantly decreased computational complexity during optimization. The inverse covariance of extended observations of the second session was only computed for the central sub-region of the electrode grid determined by the crops, such that:

$$\mathbf{C}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_2}^{-1} \in \mathbb{R}^{M_{conservative}R \times M_{conservative}R}$$

where:

$$M_{conservative} = (H - 2c_y)(W - 2c_x)$$

Since cropping reduces the number of available electrodes, the larger the crop values, the worse the decomposition outcome. Therefore, c_x and c_y were only applied during [SDA](#) optimization. Once the optimal spatial transformation was determined, minimum integer crops were recomputed ($c_{left}^*, c_{bottom}^*, c_{right}^*, c_{top}^*$) based on the [BSS](#)-optimal transformation $\mathbf{A}_{\phi^*}$.

This enabled the use of the maximum number of electrodes for the learned transformation without any out-of-bounds channels, maximizing decomposition performance post-optimization. Given the new crops, the inverse covariance matrix of extended observations was also recomputed of the second recording session for the new, larger central sub-region, where:

$$\mathbf{C}_{\tilde{\mathbf{y}}_2\tilde{\mathbf{y}}_2}^{-1} \in \mathbb{R}^{M_{max}R \times M_{max}R}$$

where:

$$M_{max} = (H - c_{top} - c_{bottom})(W - c_{left} - c_{right})$$

Loss Function

Source negentropy maximization is a common learning objective in fastICA [76], and is known to maximize source sparsity in [MU](#) decomposition [19] (See Section 2.4.4 for more information). Therefore, the loss function used in optimization was the negative of the average source negentropy approximation (J_{avg}), computed as:

$$\mathcal{L}(\mathbf{x}) = -J_{avg}(\mathbf{x}) = -\frac{1}{N} \sum_{n=1}^N J(\mathbf{x}) \quad (5.6)$$

where negentropy of a single source was approximated as:

$$J(\mathbf{x}) = (\mathbb{E}[G(\mathbf{x})] - \mathbb{E}[G(\mathbf{v})])^2 \quad (5.7)$$

In this context, $\mathbf{v}$ represents a Gaussian random variable with the same mean and variance as the random variable $\mathbf{x}$. G is the chosen contrast function, chosen in this study to be:

$$G(\mathbf{x}) = -e^{-a\frac{\mathbf{x}^2}{2}} \quad (5.8)$$

With a being a hyperparameter, chosen to be $a = 1$ as commonly chosen in previous studies. This choice of contrast function is popular in MU decomposition studies, as it is designed for robust negentropy estimation for super-Gaussian distributions [76], such as that of MU spike trains.

Optimization

The STA templates from the first session and inverse covariance matrix of extended observations from the second session are kept fixed, enabling gradients to flow through during backpropagation. As in Chapter 4, SAL applies affine transformations in a sub-differentiable manner, enabling gradient-based optimization of the full SDA pipeline. The only learnable parameters in this study were the chosen SAL affine transformation parameters. Given the gradient complexity introduced by SAL, rather than the fixed-point algorithm used in fastICA [76], first-order gradient-based methods were used for optimization.

I hypothesized that when SAL parameters were initialized far enough away from the global optimum, gradient-based optimization would not receive meaningful learning signals, being effectively stuck in a flat loss landscape, as was observed in Chapter 4. Thus, three optimization approaches were considered: gradient descent (GD), brute-force search (due to the small number of parameters), and a combination of GD and brute-force search techniques. For brute-force search, Latin Hypercube Sampling (LHS) was used, ensuring uniform coverage while keeping dimensions independent. For the combination, we used Latin Hypercube Sampling initialized gradient descent (LHS-GD), where LHS was initially used, and the optimal parameters were used as the initial conditions for GD.

Implementation Details

For gradient descent, the Adam optimizer was used with a learning rate of 0.005 and a batch size of $B = 10^4$. For LHS, one search sample referred to the negative negentropy averaged over sources ($J_{avg}(\mathbf{x})$) with all available data for a single sampled spatial transformation, ϕ_{samp} . Like for GD, the available data was also processed with a batch size of $B = 10^4$, averaging $J_{avg}(\mathbf{x})$ across batches. Since most parameters were frozen, the computational complexity of a backward pass was estimated to be approximately three times that of a forward pass. Hence, for a fair comparison, thrice as many brute-force samples were used as epochs. Thus, experiments in this study used 250 epochs for GD, 750 samples with LHS, and 375 samples with LHS followed by 125 epochs of gradient-based training LHS-GD.

Model training and evaluation were performed using an NVIDIA GeForce RTX 3080 GPU (10 GB GDDR6X memory, 8,704 CUDA cores, 272 Tensor Cores, Ampere architecture). This configuration provided sufficient computational capacity for efficient training of the PyTorch-based spatial decomposition pipeline. As in Chapter 4, SAL was implemented by leveraging the in-built PyTorch `torch.nn.functional.affine_grid` and `torch.nn.functional.grid_sample`, enabling sub-differentiable optimization.

Post-Adaptation Motor Unit Refinement

The MU decomposition pipeline in [116] used a fastICA algorithm to obtain initial source estimates followed by a convolution kernel compensation (CKC) inspired refinement procedure. Since the refinement procedure resulted in significant performance improvements, the same refinement procedure was implemented after obtaining initial source estimates from the spatial decomposition adaptation, $\hat{\mathbf{T}}(\phi^*)$. Instead of refining sources sequentially until reaching an interspike coefficient of variation (CoV) plateau, a fixed number of refinement steps N_r was defined. This fixed number allowed the simultaneous refinement of all sources with more efficient matrix operations using PyTorch. In this study, $N_r = 10$ was found to be sufficient.

Loss Landscape Sampling

To validate the use of gradient-based methods, a preliminary qualitative analysis was performed on the negentropy loss landscape. A simplified set-up where fixed transversal and/or longitudinal translation were applied at different lowpass cutoff frequencies (f_c) was used. The loss function in Equation 5.6 was then evaluated for 400 combinations of transversal/longitudinal translations bounded by $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$. This resulted in a sampled grid of the loss as a function of the parameters, which was visualized as a heatmap.

Under the assumption of high fidelity MU decomposition and manual editing, the source estimates within the same recording session should be highly non-Gaussian. Thus, for the loss landscape visualizations, a normalized negentropy measure was computed as:

$$\mathcal{L}_{norm}(\mathbf{x}) = \frac{\mathcal{L}_{intersession}(\mathbf{x})}{\mathcal{L}_{intrasession}(\mathbf{x})} \quad (5.9)$$

where a $\mathcal{L}_{norm}(\mathbf{x}) \approx 0$ implies sources are highly Gaussian relative to the original decomposition result, and $\mathcal{L}_{norm}(\mathbf{x}) \approx 1$ suggests sources are comparably non-Gaussian relative to the original decomposition. All loss landscape visualizations used $\mathcal{L}_{norm}(\mathbf{x})$ for a more intuitive scaling.

5.2.5 Simulated Spatial Transformations

To simulate the effect of electrode displacement across recording sessions, virtual spatial transformations were applied to a HD-sEMG recording with constant IED. In this study, only spatial transformations parametrized by $\phi = [T_x, T_y, \alpha]$ were considered, as translations and rotations are the only likely transformations that can realistically occur while using a rigid HD-sEMG grid. These parameters were sampled from uniform distributions constrained to $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$ and

$$\alpha \in [-15^\circ, 15^\circ].$$

To apply a sampled transformation, **SAL** with spatial parameters set to the sampled parameter values was used. Given a **HD-sEMG** grid of shape $H \times W \times T_s$, **SAL** was applied independently to every time-point, effectively operating on grayscale images of shape $H \times W$. Once applied, the expected optimal parameters in **SDA** were the same as those sampled and used in the forward transformation. The simulation outputs can be interpreted as two simultaneous recordings from two different electrode placements of the same physical process (hence, identical underlying spike trains).

5.2.6 Performance Evaluation

Rate Of Agreement

Given the discharge times of one ground truth **MU**, and the predicted discharge times for another, true positives (TP), false positives (FP) and false negatives can be estimated to compute the rate of agreement (RoA):

$$RoA = \frac{TP}{TP + FP + FN} \quad (5.10)$$

While TP corresponds to the number of correctly identified discharge times, FP corresponds to the number of erroneously predicted discharge times and FN to the number of real, undetected, discharge times. Thus, RoA successfully penalizes both discharge over and under detection. To compensate for potential temporal offsets introduced by the signal processing inherent in convolutive **BSS**, any pair of discharges that was less than 2 ms (jitter) apart was deemed a match.

Given that for both real data and phenomenologically simulated data virtual spatial transformations were applied to simulate electrode displacement, both original and transformed **HD-sEMG** data contain the same ground-truth **MUs** and discharge times. Therefore, the RoA of the predicted discharge times was computed for the transformed data by using the original **MU** discharge times as the ground truth.

Average Electrode Distance

The first step of the **SAL**-based transformation is to apply the affine transformation to the original grid coordinates. During simulated electrode displacement, since the ground truth transformation parameters were known, the degree of spatial correction post spatial adaptation was quantified. Given a simulated electrode displacement applied through affine transformation matrix $A'_{\phi_{true}}$, the pipeline estimated a spatial adaptation matrix A'_{ϕ^*} that minimizes negentropy, with the expectation that $\phi^* \approx \phi_{true}$. To compute an interpretable metric, the transformation $A'_{net} = A'_{\phi_{true}} \cdot A'^{-1}_{\phi^*}$ was applied to the original grid coordinates, and. The average Euclidean distance between original electrode positions and corresponding electrodes was computed after applying A_{net} were computed. Hence, the metric d effectively measured the extent to which the physical transformation was corrected, where if $A'_{\phi_{true}} = A'_{\phi^*}$, it follows that $d = 0$. Given a known, constant **IED**, d was computed in interpretable units.

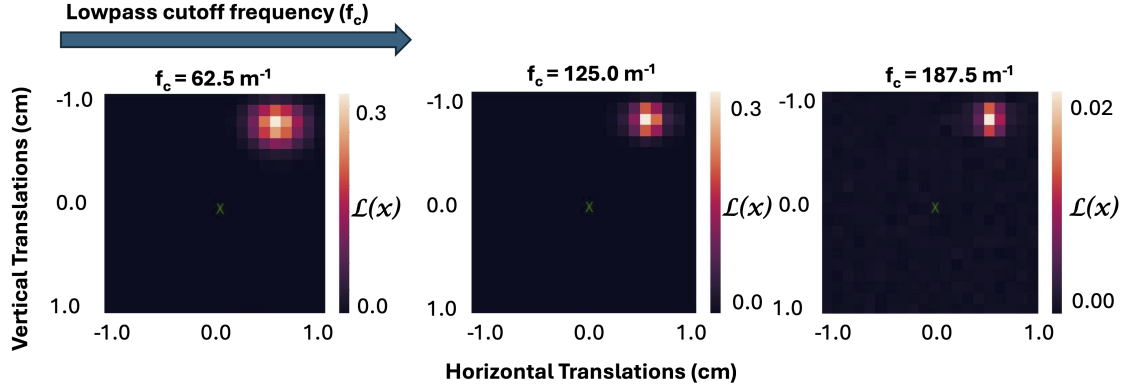


Figure 5.3: Qualitative loss landscape analysis as a function of f_c (all conditions evaluated at $SNR = 10$). Given synthetic sEMG at different f_c values, a translation of $(T_{y_0} = -0.8\text{cm}, T_{x_0} = 0.6\text{cm})$ was applied. Subsequently, the normalized negentropy ($\mathcal{L}_{norm}(\mathbf{x})$) was computed based on on this spatially transformed sEMG using the pipeline in Figure 5.2, considering 400 Spatial Adaptation Layer (SAL) corrective translations from a grid within the range $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$, resulting in the above heatmaps. Even as the spatial bandwidth increases, the blind source separation optimal region remains significantly different from other regions in the landscape, suggesting a single, robust global optimum.

5.3 Results

5.3.1 Experiment I: Mitigating Spatial Transformations With Phenomenological Simulations

Qualitative Loss Landscape Evaluation

In Figure 5.4, for different experimental conditions, the normalized negentropy ($\mathcal{L}_{norm}(\mathbf{x})$) was displayed across 400 translations, sampled at regular intervals, effectively providing a visualization of the loss as a function of spatial parameters. Given a signal-to-noise ratio (SNR) of $SNR = 10$, one normalized loss heatmap was computed for each of the spatial cutoff frequencies $f_c \in [62.50, 125.00, 187.50] m^{-1}$.

Translations across the search space only returned loss values comparable to the base loss value used for normalization in the vicinity of the optimal spatial transformation. Additionally, when the spatial lowpass cutoff frequency (f_c) was increased, the region in space that returned loss values comparable to the base loss value decreased in area. Finally, the maximum normalized negentropy ($\mathcal{L}_{norm}(\mathbf{x})$) also visibly dropped when f_c was increased.

Spatial Decomposition Adaptation Performance

For every experimental condition, a set of 50 randomly sampled spatial transformations was considered, where $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$ and $\phi \in [-15.0^\circ, 15.0^\circ]$. For each applied spatial transformation, each of the optimization methods (**LHS-GD**, **LHS**, **GD**) were applied, followed by the computation of the relevant performance metrics. Experimental conditions were considered with $SNR \in [30, 15, 5, 1] \text{ dB}$ and $f_c \in [62.50, 93.75, 125.00, 156.25, 187.50] m^{-1}$. Performance metrics were also computed before any optimization (**No Optimization**), to compare the calibration per-

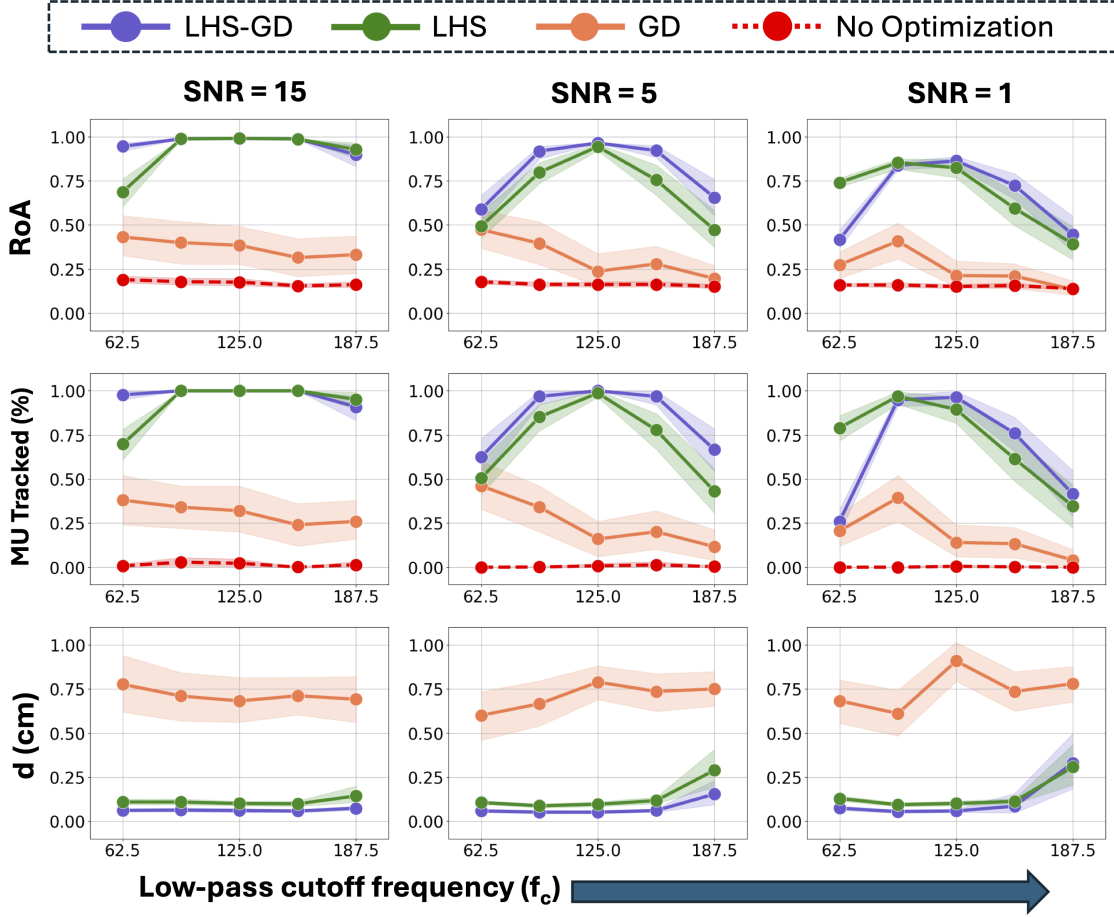


Figure 5.4: Motor unit (MU) decomposition performance metrics for Experiment I, with simulated HD-sEMG and spatial transformations. The RoA, %MU Tracked, and average electrode distance post-adaptation (d cm) were obtained for different optimization methods, signal-to-noise ratio, and spatial lowpass cutoff frequencies. While performance was usually high for mid-range values of f_c , drops in performance were observed for higher and lower values of f_c , an effect that was exacerbated by lower signal-to-noise ratio. As expected, without adaptation, the number of MUs tracked is approximately 0 across all conditions.

formance relative to naively re-using the same separation vectors and statistics from the previous recording session.

Overall Performance Comparison: As shown in Figure 5.3, **LHS-GD** and **LHS** consistently exhibited superior performance across all metrics relative to **GD** and **No Optimization**. Although the median electrode distance appeared lower for **LHS-GD** compared to **LHS** all conditions, **LHS-GD** did not exhibit visibly higher RoA and %MU Tracked across all conditions, suggesting comparable performance between methods.

Analysis of Unit Agreement and Tracking Success: While the distributions of RoA and %MU Tracked exhibited significant overlap for most conditions, for $f_c = 62.5 m^{-1}$, the lowest cutoff frequency, higher performances were observed for **LHS-GD** when SNR was high and for **LHS** when SNR was low. Additionally, while median RoA and %MU Tracked appeared to be nearly 1.00 for most conditions under $SNR = 15$ dB, (excluding $f_c = 62.5 m^{-1}$), when SNR was increased, metrics visibly decreased, an effect that was accentuated away from the central range of cutoff fre-

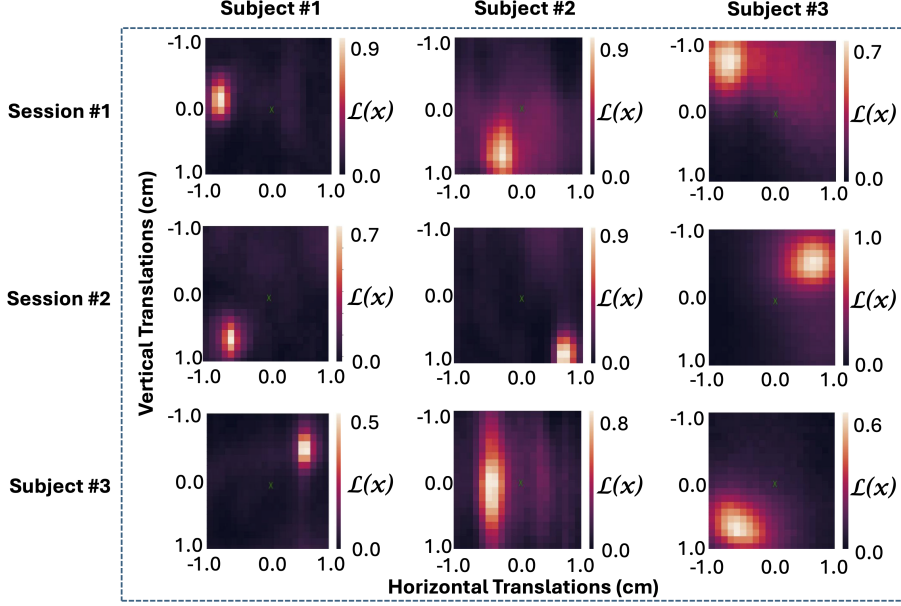


Figure 5.5: Qualitative loss landscape analysis for all subject/recording session pairs during a 25% MVC contraction. To each HD-sEMG grid, a translation sampled from a random uniform distribution was applied, where $T_{y_0}, T_{x_0} \in [-1.0, 1.0]$ cm. Subsequently, the negentropy was computed based on this spatially transformed sEMG using the pipeline in Figure 5.2, considering 400 Spatial Adaptation Layer (SAL) corrective translations from a grid within the same range as transformations were sampled from ($T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$), resulting in the above heatmaps. Sampled negentropy estimates were normalized by negentropy prior the spatial perturbation, a practical performance upper bound. In experimental recordings from healthy subjects and decomposed manually edited motor units, the blind source separation optimal region remained distinct as a robust global optimum.

quencies. Lastly, for the **No Optimization** baseline representing the use of the original separation vectors and statistics with no calibration, across all SNR and f_c conditions, %MU Tracked ≈ 0.0 .

Analysis of Spatial Correction Accuracy: **LHS-GD** and **LHS** both resulted in visibly lower median average electrode displacement post-adaptation, d (cm), relative to **GD** or **No Optimization**. In addition, lower d_{median} values were observed across conditions for **LHS-GD** relative to **LHS**. However, for an **IED** of 0.4 cm, both optimization methods resulted in $d_{median} < IED$ for all experimental conditions, and $d_{median} < \frac{IED}{2}$ for all conditions except for $f_c = 187.5\text{m}^{-1}$ and $SNR = 1\text{ dB}$.

5.3.2 Experiment II: Mitigating Simulated Spatial Transformations For Real Data

Qualitative Loss Landscape Evaluation

In Figure 5.5, the normalized negentropy ($\mathcal{L}_{norm}(\mathbf{x})$) across 400 translations sampled at regular intervals were displayed, effectively providing a visualization of the loss as a function of spatial parameters. One normalized loss heatmap was computed for each subject and recording session pair at 25% MVC. As in Figure 5.3, in Figure 5.5 loss values in the order of magnitude of the base loss used in normalisation were only noticeable in the vicinity of the spatial perturbation. This

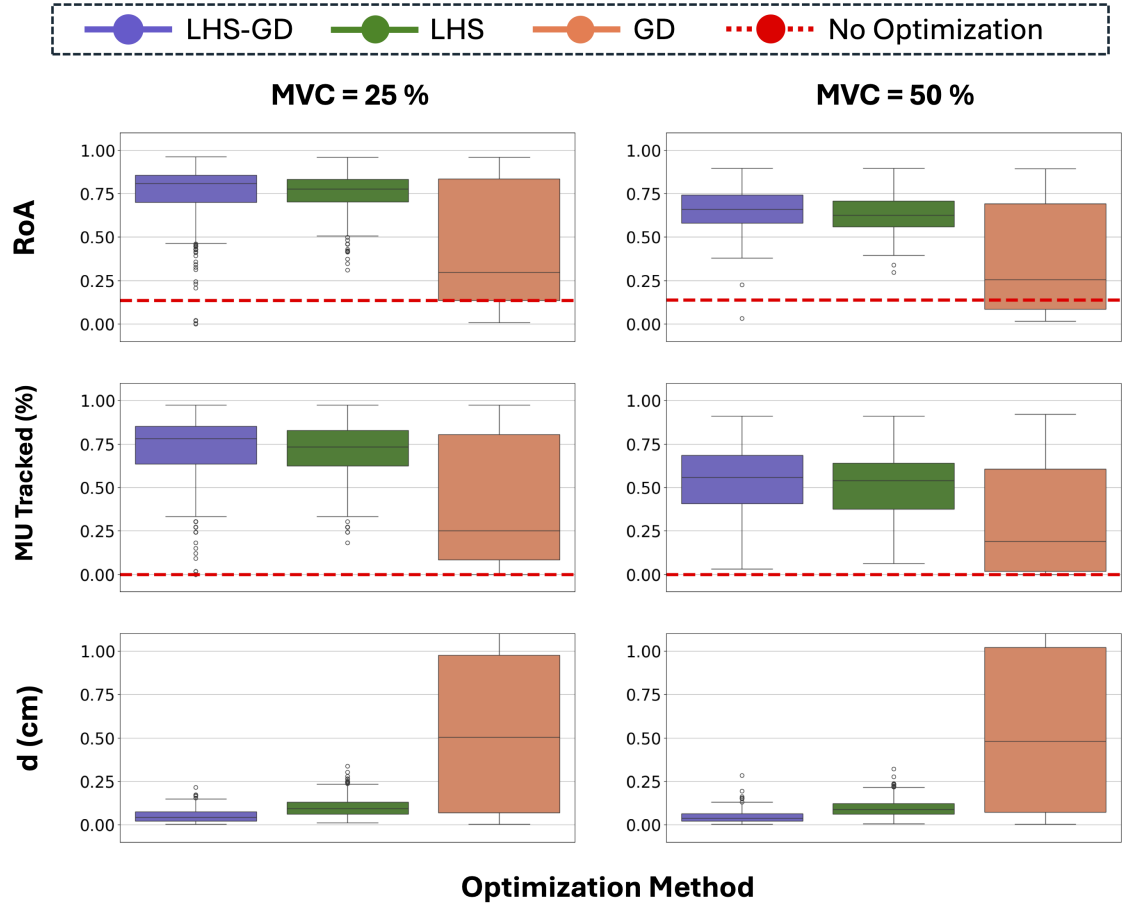


Figure 5.6: Motor unit (MU) decomposition performance metrics for Experiment II, with virtual electrode displacement being applied to real HD-sEMG and decomposed MUs. The RoA, %MU Tracked, and average electrode displacement post-adaptation, d (cm), were obtained for different optimization methods (**LHS-GD**, **LHS** and **GD**) and the two contraction levels (25% and 50%) recorded. The **No Optimization** baseline resulted in %MU tracked ≈ 0 across contraction levels. While **LHS-GD** exhibited the highest median RoA and proportion of MUs, **LHS** still exhibited a comparable performance distribution across metrics. The **GD** method exhibited a visibly larger Interquartile Range (IQR) in performance metrics relative to **LHS-GD** and **LHS**. This indicated a high trial-to-trial variability, with the performance distribution spanning almost the entire range, from complete failures (lower quartile near 0.00) to highly successful trials (upper quartile reaching ≈ 0.75).

trend was present across all subject and session pairs.

Spatial Decomposition Adaptation Performance

For every subject, contraction level, and recording session, a set of 50 randomly sampled spatial transformations was considered, where $T_x, T_y \in [-1.0\text{cm}, 1.0\text{cm}]$ and $\phi \in [-15.0^\circ, 15.0^\circ]$. For each applied spatial transformation, each of the optimization methods (**GD**, **LHS**, **LHS-GD**) were applied, followed by the computation of the relevant performance metrics. Performance metrics before any optimization (**No Optimization**) were also computed to compare the calibration performance relative to naively re-using the same separation vectors and statistics from the previous session.

Overall Performance Comparison: As shown in Figure 5.6, **LHS-GD** and **LHS** consistently exhibited superior performance across all metrics relative to **GD** and **No Optimization**. Although the median average electrode distance appeared visibly lower for **LHS-GD** compared to **LHS**, there was a large distribution overlap between the two methods for the RoA and **MU**-tracking distributions, suggesting comparable performance between the methods.

Analysis of Unit Agreement and Tracking Success: The RoA and %MU Tracked distributions exhibited a dependence on contraction intensity across methods. While **LHS-GD** and **LHS** achieved robust RoA (median ≈ 0.80) and %MU tracked (median ≈ 0.75) during a 25% **MVC** contraction, these median performances visibly dropped with an increase in contraction level (25% to 50% **MVC**). Furthermore, the outlier markers (metric $> 1.5 * IQR$) in the boxplots from Figure 5.6 appeared to be visibly lower in RoA and %MU Tracked for **LHS-GD** relative to **LHS**. Additionally, the **GD** method shows a visibly larger Interquartile Range (IQR) in performance metrics relative to **LHS-GD** and **LHS**. This indicated a high trial-to-trial variability, with the performance distribution spanning almost the entire range, from complete failures (lower quartile near 0.00) to highly successful trials (upper quartile reaching ≈ 0.75). Furthermore, the **No Optimization** baseline, representing the use of the original separation vectors and statistics for decomposition, consistently resulted in near zero **MUs** tracked.

Analysis of Spatial Correction Accuracy: For both **LHS-GD** and **LHS**, across all contraction levels, substantially lower median and IQR average electrode distance were obtained relative to **GD** and **No Optimization**. Furthermore, both methods resulted in physically low median distance values relative to the **IED**, such that $d_{median} < \frac{IED}{2}$ m where $IED = 0.4$ cm. In contrast, the distribution of d when using **GD** was comparable to the median simulated electrode displacement distance applied to the original **HD-sEMG** grid.

5.4 Discussion

5.4.1 Loss Landscape Properties & Optimization

The qualitative analysis of the loss landscape, performed on both simulated (Experiment I) and real data (Experiment II), demonstrates that **BSS**-optimality (source sparsity) is a feasible learning objective for spatially adapting the separation vectors to mitigating electrode displacement. As illustrated across conditions in Figures 5.4 and for all subjects and sessions in 5.5, the loss landscape was characterized by a "flat" topology except for a distinct region in the vicinity of the single, bowl-shaped minimum, suggesting a convex optimization problem. However, this topography also highlights a critical optimization challenge: the vast majority of the search space provides no gradient information, rendering purely gradient-based methods ineffective without a specific initialization strategy.

Based on loss landscape visualizations, increasing the spatial lowpass cutoff frequency (f_c) caused the basin of attraction to decrease in area. This was likely caused by higher-frequency components decreasing the spatial autocorrelation more rapidly with translation magnitude, sharpening the bowl-shaped region and making the landscape harder to traverse via gradient descent. Ad-

ditionally, the observed drop in maximum normalized negentropy with increasing f_c suggests an increase in bilinear interpolation error associated with higher frequency content, resulting in greater sensitivity to noise. Despite these frequency-dependent challenges, the loss landscape retained the visual loss landscape properties that suggest optimizability. This was further validated by the presence of real loss landscapes in Figure 5.5, supporting the optimisability of the spatial adaptation problem even for real-world EMG spatial frequency characteristics.

5.4.2 Optimization Strategy and Performance

Consistent with the landscape topography described above, the performance metrics in both Experiment I (Figure 5.4) and Experiment II (Figure 5.6) confirm that brute-force initialization is a prerequisite for successful adaptation. Both **LHS** and **LHS-GD** significantly outperformed the use of **GD** alone, which frequently resulted in near-zero unit tracking, likely due to poor initialization on the flat areas of the loss landscape.

While for both Experiment I and Experiment II, the **LHS-GD** method consistently achieved lower median electrode displacement (d) post-adaptation relative to **LHS**, **LHS-GD** and **LHS** yielded comparable decomposition metrics (RoA and %MU Tracked) across most experimental conditions and recording session and subject pairs. Interestingly, this improvement in spatial accuracy did not always translate to tangibly higher decomposition performance. This suggests that perfect spatial adaptation is not strictly necessary. As long as the estimated parameters are within a proximity threshold of the optimal spatial correction, the decomposition algorithm can successfully recover the **MUs**.

Additionally, there was an observed trade-off regarding robustness. In Experiment I, while **LHS-GD** generally performed well, **LHS** exhibited superior performance in specific challenging conditions (e.g., low SNR). This performance gap is likely attributable to the initialization density: **LHS-GD** utilizes fewer brute-force search points than pure **LHS** for comparable computational complexity. Consequently, in scenarios with high noise (low SNR), **LHS-GD** is statistically more likely to miss the global optimum’s basin during initialization, leading to the worse failure modes observed in the outlier analysis. Therefore, due to benefits in robustness, **LHS** is most likely to provide robust **MU** tracking across conditions. Since this method is only dependent on a brute-force search algorithm, it can also more readily be implemented within existing **MU** decomposition software and packages.

5.4.3 Physical Interpretation of Adaptation

Finally, the spatial correction accuracy observed across both experiments implies a high degree of physical precision in the adaptation. For the majority of experimental conditions, both **LHS** and **LHS-GD** resulted in a median electrode displacement of $d_{median} < \frac{IED}{2}$. This indicates that both **LHS** and **LHS-GD** successfully corrected the spatial perturbation to within less than half of the **IED**, effectively mitigating the effects of electrode shift to a sub-electrode level. Thus, not only does spatial decomposition adaptation provide robust **MU** tracking, but it also provides a reliable estimate of the original electrode displacement, an interpretable measure of the physical origins of the intersession variability.

5.5 Conclusion

In this study, I re-frame the problem of **MU** tracking in isometric contractions from template matching **MUAPs** of independently decomposed recording sessions to adapting separation vectors from session so they can be re-used in another. I propose and validate a spatial decomposition adaptation framework that utilizes **SAL** (introduced in Chapter 4) to spatially transform separation vectors, allowing for their re-use in another recording session for the same contraction level.

Through phenomenological simulations and virtual displacements applied to real **HD-sEMG** data, I analysed the loss landscape of a **BSS**-inspired loss function. While the optimization problem is convex with a robust global optimum, the landscape is predominantly flat, rendering gradient-based methods ineffective without proper initialization. Consequently, the implemented hybrid approach combining brute-force search and gradient descent (**LHS-GD**), or even sufficiently dense sampling (**LHS**) alone, consistently outperform standard gradient descent. These methods achieved high RoA and result in the successful tracking of a significant proportion of **MUs**, correcting spatial displacements to sub-electrode levels of precision (often less than half the **IED**).

The implications of these findings are significant for the robustness of **MU** decomposition and longitudinal tracking. The need of precise physical markings on the skin is bypassed by mitigating the physical effects introduced by electrode displacement. Crucially, the introduced method successfully meets the motor unit tracking success criterion defined in Chapter 1 of %MU Tracked $\geq 40\%$, albeit in simulated electrode displacement conditions only.

Future work will focus on validating this framework on a larger dataset collected across multiple physical recording sessions to confirm its efficacy under complex, real-world conditions. To facilitate widespread adoption in the neurophysiology community, I aim to implement the brute-force search only variation of this pipeline as an open-source package in both Matlab and Python (NumPy). Additionally, given the distinct topology of the loss landscape, I plan to explore non-gradient optimization methods, such as Bayesian optimization, to improve convergence efficiency.

Finally, beyond motor unit tracking, this adaptation framework holds promise as an **unsupervised learning** objective for estimating physically relevant quantities. For instance, since **SDA** resulted in more accurate spatial parameter estimation relative to **SAL**-based adaptation results for gesture classification reported in Chapter 4, **SDA** may be used as an unsupervised proxy task to mitigate the effects of electrode displacement for gesture recognition classifiers, joint angle regressors, among other **HD-sEMG**-based machine learning models. More generally, **BSS**-inspired objectives can be extended to the estimation of other meaningful quantities. Recently, this **BSS**-inspired optimization approach has been extended to the estimate of volume conduction properties (e.g. fibre conduction velocity) based on biophysical **HD-sEMG** models [62]³. Thus, future work will investigate the use of this framework in other applications as an inverse modelling tool.

³As a co-author of the referenced study, my contribution was the **BSS**-inspired objective into the introduced pipeline.

Chapter 6

General Discussion & Conclusions

6.1 General Summary

The aim of this thesis was to develop machine learning approaches to improve the intersession generalization of [surface electromyography \(sEMG\)](#) motor intent estimation without resource-intensive data collection efforts or impractically strenuous calibration sessions. A novel spatial data augmentation was introduced in Chapter 3 to probe models to learn electrode-placement invariant representations, with the goal of zero-shot intersession generalization. However, this approach did not significantly outperform dimensionality reduction based regularization, and augmentation-enhanced performance remained low.

To address these limitations, the [Spatial Adaptation Layer \(SAL\)](#) was introduced in Chapter 4 to mitigate electrode displacement in a few-shot, input-based domain adaptation framework. [SAL](#) achieved significant performance improvements relative to implemented state-of-the-art adaptation methods when using a subset of available [gestures](#)¹. To demonstrate the versatility of [SAL](#), it was subsequently employed to mitigate electrode displacement based on [motor unit \(MU\)](#) decomposition models in Chapter 5. In this context, [SAL](#) was effectively able to mitigate electrode shift, matching the majority of decomposed [MUs](#) with high rates of agreement. Collectively, this body of work establishes a new methodological framework for mitigating the effects of electrode displacement, synthesized and evaluated in the following sections.

6.2 Key Findings

6.2.1 Spatial Data Augmentation

Spatial data augmentation was initially hypothesized to probe models to learn electrode-placement invariant representations. This was approached by treating different channel rows in a 2D [high-density surface electromyography \(HD-sEMG\)](#) system as distinct recording sessions, thereby increasing the effective training set size at the expense of spatial coverage. While intersession ac-

¹Throughout this thesis, the standard term [gesture](#) is used as a macro-level class label for configurations of sustained isometric muscle effort rather than physical movement. See the front-matter Glossary for the complete methodological definition.

curacy significantly improved relative to an unregularized baseline, crucially, these augmentation-enhanced performances did not significantly outperform principal component analysis (PCA) dimensionality reduction.

This equivalence suggests that the spatial augmentation functions primarily as a form of implicit model regularization rather than successfully encoding the geometric structure of the shift. This finding was consistent across all considered feature sets, some of which outperformed the time-domain [convolutional neural network \(CNN\)](#) implemented. This indicates that the limitation is intrinsic to the data strategy rather than the model architecture. Ultimately, across all implemented methods, final intersession accuracies remained low ($< 50\%$), suggesting that generic spatial data augmentation alone is insufficient to bridge the domain gap required for robust intersession generalization.

Limitations: Ultimately, the extent of spatial transformation invariance introduced by data augmentation is bounded by the variability of transformations introduced and the capacity of the model to learn invariant representations. In particular, the method investigated was restricted to proximal-distal shifts, as preliminary experiments indicated that training on more complex transformations yielded worse intersession performance. However, this was not evaluated further with more expressive models than the lightweight temporal [CNN](#) implemented. Furthermore, the high variability in performance improvement across subjects suggests that other confounding factors such as limb position, contraction intensity, and gesture technique may outweigh the benefits of spatial augmentation. Consequently, while augmentation provides a useful regularization effect, it does not yield in session-invariant representations across an array of different machine learning models.

6.2.2 Spatial Input-Based Domain Adaptation

Given low reported performance for spatial data augmentation, I investigated the use of [SAL](#): a layer that learns a single optimal spatial transformation to mitigate the effects of electrode displacement with respect to the original recording session.

Gesture Classification: [SAL](#)-based performances challenge the reliance on complex model architectures for intersession generalization. Crucially, a [SAL](#)-adapted logistic regressor significantly outperformed an established [CNN](#) architecture using alternative domain adaptation methods. This dominance of a linear classifier suggests that the bottleneck in [HD-sEMG](#) generalization is not the separability of the features, but the differences between source and target recording sessions. With an adaptation set containing only three of six total gestures, [SAL](#) achieved accuracies of $\geq 85\%$ on regular-array [HD-sEMG](#) recordings, approximately reverting the intersession problem to one of intrasession generalization. This confirms that by resolving the physical transformation based on inductive biases, simpler models are sufficient to achieve high-performance classification.

Motor Unit Tracking: The application of [SAL](#) to [MU](#) tracking serves as a critical validation of the method's ability to directly mitigate the physical effects of electrode displacement. When tracking [MUs](#) in experimental [HD-sEMG](#) recordings with simulated spatial transformations, the baseline condition (no adaptation) tracked $\approx 0\%$ of [MUs](#), highlighting the fragility of tracking

approaches such as [100] under realistic electrode displacements. On the other hand, **SAL** successfully mitigated these effects across all contraction levels. By adapting the separation vectors directly, **SAL** reframes the challenge of **MU** tracking from a rigid template-matching problem to a domain-adaptation problem. Consequently, this success suggests that **SAL** is not limited to classification tasks, but can be generally applied to mitigate electrode displacements given any task-relevant loss function.

Interpretability & Optimization: Beyond effectively mitigating the effects of electrode displacement, **SAL** directly estimates physically meaningful **parameters** in an interpretable way. The validity of these parameters was confirmed by the significant linear correlation between accuracy improvements and transformation magnitude, specifically translation across the length of muscle fibres, which aligns with past physiological studies [63, 1]. This physical fidelity resulted in a significant reduction of average electrode displacement in the majority of trials. This was especially pronounced in the **MU** tracking context, where the average post-adaptation displacement was reduced to ≈ 0.2 cm.

Furthermore, I found that due to the small number of **SAL** parameters, brute-force optimization yielded similar performances to gradient-based approaches with comparable computational complexity for **MU** tracking and gesture classification. This suggests that **SAL**-based adaptation can be readily implemented without the overhead of nuanced automatic differentiation frameworks, such as *PyTorch* or *TensorFlow*. Thus, **SAL** offers a dual advantage: it provides the robustness of physically grounded adaptation while remaining computationally lightweight enough for deployment in resource-constrained embedded environments.

Limitations: While **SAL** offers a robust solution for regular arrays, its efficacy is dependent on the assumption of a constant **interelectrode distance (IED)**. Consequently, performance on irregular grids was notably inferior to that on regular arrays. Furthermore, since **SAL** exclusively targets electrode displacement, it requires additional preprocessing methods such as including baseline noise reduction and spatial filtering to achieve optimal performances. Finally, a critical limitation of the **MU** tracking validation is its reliance on simulated spatial transformations. Thus, the framework has yet to be validated on longitudinal experimental recordings involving physical electrode re-placement across multiple days.

6.3 Implications

6.3.1 Leveraging Inductive Biases Of Intersession Variability

The most significant theoretical implication of this work is that electrode displacement should be modelled as a structured geometric transformation rather than generic statistical noise. While complex, deep learning models have successfully learned transformation-invariant representations across domains, the number of subjects required for intersubject generalization is impractical for numerous downstream **HD-sEMG**-based applications. Similarly, while model-based adaptation approaches significantly improve intersession generalization, the necessity of label-balance in the adaptation set can lead to impractically long calibration procedures. Alternatively, by leveraging

inductive biases of the physical transformations that occur to HD-sEMG signals across sessions, intersession generalization can be significantly improved with small calibration sets, a promising finding for downstream HD-sEMG-based applications. Crucially, since electrode displacement naturally occurs in any application that requires re-placement during use, spatial input-based adaptation can be generally applied to any HD-sEMG-based machine learning task, as shown with gesture classification and MU tracking.

6.3.2 Input-Based Adaptation: From Intersession To Intrasession

One interpretation of input-based adaptation is that, by minimizing the effects of intersession transformations on the target domain, the problem of intersession generalization is effectively reduced to one of intrasession generalization. Within sessions, numerous HD-sEMG-based machine learning models have achieved high performances, even with simple feature sets and linear classifiers. Thus, spatial input-based adaptation enables the robust use of more interpretable lightweight models, as shown in Chapter 4 by the high intersession accuracy achieved with a single feature per channel and a logistic regressor. Furthermore, the use of lightweight models and sufficiency of brute-force optimization can facilitate the implementation of HD-sEMG-based machine learning models in embedded systems.

6.3.3 Motor Unit Tracking: An Input-Based Adaptation Problem

Currently, MU decomposition is predominantly achieved through linear convolutive blind source separation (BSS)), which provide interpretable results and has the guarantee of source identifiability [70, 116]. Since linear models are not expressive enough to learn invariant representations, input-based domain adaptation is the most suitable method for preserving original decomposition filters and identifiability. While spatial decomposition adaptation has not been validated with experimental signals from multi-day recordings, it successfully mitigated the effects of simulated spatial transformations and longitudinal MU tracking has been successfully achieved when controlling for electrode displacement. The successful tracking of MUs across sessions in real-world settings enables various longitudinal applications, such as single MU control in spinal cord injury patients [119], as well as monitoring disease progression in neuromuscular diseases such as ALS [153].

6.4 Future Outlook

6.4.1 Algorithmic Extensions

While brute-force optimization was sufficient across machine learning tasks to learn corrective spatial transformations, more sample efficient non-gradient methods can be implemented to tackle the SAL loss landscape. Thus, future work will also consider black-box optimization methods such as Bayesian and particle swarm optimization. Furthermore, the use of SAL is currently restricted to HD-sEMG arrays with constant IED. This will be extended to handle non-uniform arrays with known electrode spacing. Finally, since non-gradient methods were successfully applied, future

work will spatial input-based adaptation will be applied to other popular, non-differentiable traditional classifiers, such as random forests or support vector machines.

6.4.2 Motor Unit Tracking: Experimental & Clinical Validation

Beyond the successful mitigation of the effects of simulated spatial transformation, future work will investigate the application of spatial decomposition adaptation to experimental, longitudinal **HD-sEMG** recordings, which contains a more realistic range of non-stationarities. Once validated in healthy subjects, this method will also be extended further to single **MU** control and clinical assessments in patients with motor impairments.

6.4.3 Decomposition As A Universal Calibration Proxy

Interestingly, spatial decomposition adaptation resulted in a more significant reduction of simulated spatial transformations relative to the use of a gesture classification learning objective. This is likely due to the degree of information encoded in the spatiotemporal **motor unit action potential (MUAP)** fingerprints involved, as well as its direct association with physical, objective quantities. Therefore, future work will investigate spatial decomposition adaptation based on a single isometric contraction as a universal proxy learning objective for intersession generalization in downstream tasks.

6.4.4 Biophysics-Informed Inverse Modelling

Beyond spatial decomposition adaptation, I more generally introduce a framework for using **BSS**-inspired loss functions for estimating specific quantities of interest. In spatial decomposition adaptation, the spatial transformation parameters that maximize the sparsity of predicted sources are estimated. Recently, this framework was applied to an **HD-sEMG** biophysical forward model. The biophysics model generates realistic **MUAPs** based on a low-dimensional set of physical and anatomical parameters (e.g. fibre conduction velocity) [62]. In this context, the simulation parameters were optimized by constructing separation vectors from generated **MUAPs** and maximizing the output source sparsity. This not only enabled robust, biophysically grounded decomposition, but also directly estimated physically meaningful quantities. Thus, future work will expand on biophysically-inspired decomposition, as well as investigate the use of this framework in other applications as an inverse modelling tool.

Bibliography

- [1] Christoph Amma et al. “Advancing Muscle-Computer Interfaces with High-Density Electromyography”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 929–938. ISBN: 9781450331456. DOI: [10.1145/2702123.2702501](https://doi.org/10.1145/2702123.2702501). URL: <https://doi.org/10.1145/2702123.2702501>.
- [2] A. Anastasiev et al. “Automated EMG-Based Classification of Upper Extremity Motor Impairment Levels in Subacute Stroke”. In: *Sensors (Basel, Switzerland)* 25.22 (2025), p. 6829. DOI: [10.3390/s25226829](https://doi.org/10.3390/s25226829). URL: <https://doi.org/10.3390/s25226829>.
- [3] David Arthur and Sergei Vassilvitskii. “k-means++: the advantages of careful seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245.
- [4] Manfredo Atzori, Matteo Cognolato, and Henning Müller. “Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands”. In: *Frontiers in Neurorobotics* Volume 10 - 2016 (2016). ISSN: 1662-5218. DOI: [10.3389/fnbot.2016.00009](https://doi.org/10.3389/fnbot.2016.00009). URL: <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2016.00009>.
- [5] Manfredo Atzori et al. “Electromyography data for non-invasive naturally-controlled robotic hand prostheses”. In: *Scientific Data* 1.1 (Dec. 2014), p. 140053. ISSN: 2052-4463. DOI: [10.1038/sdata.2014.53](https://doi.org/10.1038/sdata.2014.53). URL: <https://doi.org/10.1038/sdata.2014.53>.
- [6] Manfredo Atzori et al. “Electromyography data for non-invasive naturally-controlled robotic hand prostheses”. In: *Scientific Data* 1.1 (2014), pp. 1–13. DOI: [10.1038/sdata.2014.53](https://doi.org/10.1038/sdata.2014.53). URL: <https://doi.org/10.1038/sdata.2014.53>.
- [7] Simon Avrillon et al. “The decoding of extensive samples of motor units in human muscles reveals the rate coding of entire motoneuron pools”. In: *bioRxiv* (2023). DOI: [10.1101/2023.11.25.568607](https://doi.org/10.1101/2023.11.25.568607). eprint: <https://www.biorxiv.org/content/early/2023/11/25/2023.11.25.568607.full.pdf>. URL: <https://www.biorxiv.org/content/early/2023/11/25/2023.11.25.568607>.
- [8] Arjun K. Bansal et al. “Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials”. In: *Journal of Neurophysiology* 107.5 (2012). PMID: 22157115, pp. 1337–1355. DOI: [10.1152/jn.00781.2011](https://doi.org/10.1152/jn.00781.2011). eprint: <https://doi.org/10.1152/jn.00781.2011>. URL: <https://doi.org/10.1152/jn.00781.2011>.
- [9] Deren Y. Barsakcioglu et al. “Control of Spinal Motoneurons by Feedback From a Non-Invasive Real-Time Interface”. In: *IEEE Transactions on Biomedical Engineering* 68.3 (2021), pp. 926–935. DOI: [10.1109/TBME.2020.3001942](https://doi.org/10.1109/TBME.2020.3001942).

- [10] Mikhail Belkin and Partha Niyogi. “Laplacian Eigenmaps for dimensionality reduction and data representation”. In: *Neural Comput.* 15.6 (June 2003), pp. 1373–1396. ISSN: 0899-7667. DOI: [10.1162/089976603321780317](https://doi.org/10.1162/089976603321780317). URL: <https://doi.org/10.1162/089976603321780317>.
- [11] Shai Ben-David et al. “A theory of learning from different domains”. In: *Machine Learning* 79 (2010), pp. 151–175. URL: <https://api.semanticscholar.org/CorpusID:8577357>.
- [12] Denise J. Berger and Andrea d’Avella. “Effective force control by muscle synergies”. In: *Frontiers in Computational Neuroscience* Volume 8 - 2014 (2014). ISSN: 1662-5188. DOI: [10.3389/fncom.2014.00046](https://doi.org/10.3389/fncom.2014.00046). URL: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2014.00046>.
- [13] Niels Birbaumer, Sergio Ruiz, and Ranganatha Sitaram. “Learned regulation of brain metabolism”. In: *Trends in Cognitive Sciences* 17.6 (June 2013), pp. 295–302. DOI: [10.1016/j.tics.2013.04.009](https://doi.org/10.1016/j.tics.2013.04.009). URL: <https://doi.org/10.1016/j.tics.2013.04.009>.
- [14] Nathan E. Bunderson and Todd A. Kuiken. “Quantification of Feature Space Changes With Experience During Electromyogram Pattern Recognition Control”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 20.3 (2012), pp. 239–246. DOI: [10.1109/TNSRE.2011.2182525](https://doi.org/10.1109/TNSRE.2011.2182525).
- [15] Arnault H. Caillet et al. “Larger and Denser: An Optimal Design for Surface Grids of EMG Electrodes to Identify Greater and More Representative Samples of Motor Units”. In: *eNeuro* 10.9 (2023). DOI: [10.1523/ENEURO.0064-23.2023](https://doi.org/10.1523/ENEURO.0064-23.2023). eprint: <https://www.eneuro.org/content/10/9/ENEURO.0064-23.2023.full.pdf>. URL: <https://www.eneuro.org/content/10/9/ENEURO.0064-23.2023>.
- [16] Evan Campbell, Angkoon Phinyomark, and Erik Scheme. “Current Trends and Confounding Factors in Myoelectric Control: Limb Position and Contraction Intensity”. In: *Sensors* 20.6 (2020). ISSN: 1424-8220. DOI: [10.3390/s20061613](https://doi.org/10.3390/s20061613). URL: <https://www.mdpi.com/1424-8220/20/6/1613>.
- [17] Ujwal Chaudhary, Niels Birbaumer, and António Ramos-Murguialday. “Brain–computer interfaces for communication and rehabilitation”. In: *Nature Reviews Neurology* 12 (2016), pp. 513–525. DOI: [10.1038/nrneurol.2016.113](https://doi.org/10.1038/nrneurol.2016.113). URL: <https://doi.org/10.1038/nrneurol.2016.113>.
- [18] Jiangcheng Chen et al. “High-Density Surface EMG-Based Gesture Recognition Using a 3D Convolutional Neural Network”. In: *Sensors* 20.4 (2020). ISSN: 1424-8220. DOI: [10.3390/s20041201](https://doi.org/10.3390/s20041201). URL: <https://www.mdpi.com/1424-8220/20/4/1201>.
- [19] Maoqi Chen and Ping Zhou. “A Novel Framework Based on FastICA for High Density Surface EMG Decomposition”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.1 (2016), pp. 117–127. DOI: [10.1109/TNSRE.2015.2412038](https://doi.org/10.1109/TNSRE.2015.2412038).
- [20] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD ’16*. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 785–794. ISBN: 9781450342322. DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785). URL: <https://doi.org/10.1145/2939672.2939785>.
- [21] Wei Chen et al. “An Extended Spatial Transformer Convolutional Neural Network for Gesture Recognition and Self-Calibration Based on Sparse sEMG Electrodes”. In: *IEEE Transactions on Biomedical Circuits and Systems* 16.6 (2022), pp. 1204–1215. DOI: [10.1109/TBCAS.2022.3222196](https://doi.org/10.1109/TBCAS.2022.3222196).
- [22] Vincent C. K. Cheung et al. “Muscle synergy patterns as physiological markers of motor cortical damage”. In: *Proceedings of the National Academy of Sciences* 109.36 (2012), pp. 14652–14656. DOI: [10.1073/pnas.1212056109](https://doi.org/10.1073/pnas.1212056109). URL: <https://doi.org/10.1073/pnas.1212056109>.

- [23] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *CoRR* abs/1406.1078 (2014). arXiv: [1406.1078](https://arxiv.org/abs/1406.1078). URL: <http://arxiv.org/abs/1406.1078>.
- [24] Dan Claudiu Ciresan et al. “Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition”. In: *CoRR* abs/1003.0358 (2010). arXiv: [1003.0358](https://arxiv.org/abs/1003.0358). URL: <http://arxiv.org/abs/1003.0358>.
- [25] Alexander Kenneth Clarke et al. “Deep Learning for Robust Decomposition of High-Density Surface EMG Signals”. In: *IEEE Transactions on Biomedical Engineering* 68.2 (2021), pp. 526–534. DOI: [10.1109/TBME.2020.3006508](https://doi.org/10.1109/TBME.2020.3006508).
- [26] Ulysse Côté-Allard et al. “A Transferable Adaptive Domain Adversarial Neural Network for Virtual Reality Augmented EMG-Based Gesture Recognition”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 546–555. DOI: [10.1109/TNSRE.2021.3059741](https://doi.org/10.1109/TNSRE.2021.3059741).
- [27] Ulysse Côté-Allard et al. “Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.4 (2019), pp. 760–771. DOI: [10.1109/TNSRE.2019.2896269](https://doi.org/10.1109/TNSRE.2019.2896269).
- [28] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177).
- [29] Carlo J. De Luca and Roberto Merletti. “Surface myoelectric signal cross-talk among muscles of the leg”. In: *Electroencephalography and clinical neurophysiology* 69.6 (1988), pp. 568–575. DOI: [10.1016/0013-4694\(88\)90169-1](https://doi.org/10.1016/0013-4694(88)90169-1). URL: [https://doi.org/10.1016/0013-4694\(88\)90169-1](https://doi.org/10.1016/0013-4694(88)90169-1).
- [30] Carlo J. De Luca et al. “Decomposition of Surface EMG Signals”. In: *Journal of Neurophysiology* 96.3 (2006). PMID: 16899649, pp. 1646–1657. DOI: [10.1152/jn.00009.2006](https://doi.org/10.1152/jn.00009.2006). eprint: <https://doi.org/10.1152/jn.00009.2006>. URL: <https://doi.org/10.1152/jn.00009.2006>.
- [31] Carlo J. De Luca et al. “Filtering the surface EMG signal: Movement artifact and baseline noise contamination”. In: *Journal of Biomechanics* 43.8 (2010), pp. 1573–1579. DOI: [10.1016/j.jbiomech.2010.01.027](https://doi.org/10.1016/j.jbiomech.2010.01.027).
- [32] Marc Peter Deisenroth, Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020. ISBN: 9781108455145. URL: <https://www.cambridge.org/au/universitypress/subjects/computer-science/pattern-recognition-and-machine-learning/mathematics-machine-learning?format=PB>.
- [33] A. Del Vecchio et al. “Tutorial: Analysis of motor unit discharge characteristics from high-density surface EMG signals”. In: *Journal of Electromyography and Kinesiology* 53 (2020), p. 102426. ISSN: 1050-6411. DOI: <https://doi.org/10.1016/j.jelekin.2020.102426>. URL: <https://www.sciencedirect.com/science/article/pii/S1050641120300419>.
- [34] Jorrit P. van Dijk et al. “Evidence of Potential Averaging over the Finite Surface of a Bioelectric Surface Electrode”. In: *Annals of Biomedical Engineering* 37 (2009), pp. 1141–1151. DOI: [10.1007/s10439-009-9680-7](https://doi.org/10.1007/s10439-009-9680-7).
- [35] G. V. Dimitrov and N. A. Dimitrova. “Precise and fast calculation of the motor unit potentials detected by a point and rectangular plate electrode”. In: *Medical Engineering & Physics* 20.5 (1998), pp. 374–381. DOI: [10.1016/S1350-4533\(98\)00014-9](https://doi.org/10.1016/S1350-4533(98)00014-9). URL: [https://doi.org/10.1016/S1350-4533\(98\)00014-9](https://doi.org/10.1016/S1350-4533(98)00014-9).
- [36] William B. Dolan and Chris Brockett. “Automatically Constructing a Corpus of Sentential Paraphrases”. In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. 2005. URL: <https://aclanthology.org/I05-5002/>.

- [37] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *CoRR* abs/2010.11929 (2020). arXiv: [2010.11929](https://arxiv.org/abs/2010.11929). URL: <https://arxiv.org/abs/2010.11929>.
- [38] Yu Du et al. “Surface EMG-Based Inter-Session Gesture Recognition Enhanced by Deep Domain Adaptation”. In: *Sensors* 17.3 (2017). ISSN: 1424-8220. DOI: [10.3390/s17030458](https://doi.org/10.3390/s17030458). URL: <https://www.mdpi.com/1424-8220/17/3/458>.
- [39] K. Englehart and B. Hudgins. “A robust, real-time control scheme for multifunction myoelectric control”. In: *IEEE Transactions on Biomedical Engineering* 50.7 (2003), pp. 848–854. DOI: [10.1109/TBME.2003.813539](https://doi.org/10.1109/TBME.2003.813539).
- [40] K. Englehart et al. “Classification of the myoelectric signal using time-frequency based representations”. In: *Medical & Biological Engineering & Computing* 21.6-7 (1999), pp. 431–438.
- [41] R. M. Enoka and J. Duchateau. “Physiology of Muscle Activation and Force Generation”. In: *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd, 2016. Chap. 1, pp. 1–29. ISBN: 9781119082934. DOI: <https://doi.org/10.1002/9781119082934.ch01>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119082934.ch01>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119082934.ch01>.
- [42] Roger M. Enoka. *Neuromechanics of Human Movement*. 6th ed. Champaign, IL: Human Kinetics, 2024. ISBN: 978-1718213739.
- [43] Paria Esmatloo et al. “Dynamic Finger Task Identification Using Electromyography”. In: *2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechanics (BioRob)*. 2022, pp. 1–6. DOI: [10.1109/BioRob52689.2022.9925498](https://doi.org/10.1109/BioRob52689.2022.9925498).
- [44] Xinchun Fan et al. “CSAC-Net: Fast Adaptive sEMG Recognition through Attention Convolution Network and Model-Agnostic Meta-Learning”. In: *Sensors* 22.10 (2022). ISSN: 1424-8220. DOI: [10.3390/s22103661](https://doi.org/10.3390/s22103661). URL: <https://www.mdpi.com/1424-8220/22/10/3661>.
- [45] D. Farina, D. F. Stegeman, and R. Merletti. “Biophysics of the Generation of EMG Signals”. In: *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd, 2016. Chap. 2, pp. 1–24. ISBN: 9781119082934. DOI: <https://doi.org/10.1002/9781119082934.ch02>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119082934.ch02>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119082934.ch02>.
- [46] D. Farina et al. “A surface EMG generation model with multilayer cylindrical description of the volume conductor”. In: *IEEE Transactions on Biomedical Engineering* 51.3 (2004), pp. 415–426. DOI: [10.1109/TBME.2003.820998](https://doi.org/10.1109/TBME.2003.820998).
- [47] D. Farina et al. “Selectivity of spatial filters for surface EMG detection from the tibialis anterior muscle”. In: *IEEE Transactions on Biomedical Engineering* 50.3 (2003), pp. 354–364. DOI: [10.1109/TBME.2003.808830](https://doi.org/10.1109/TBME.2003.808830).
- [48] Dario Farina and Aleš Holobar. “Characterization of Human Motor Units From Surface EMG Decomposition”. In: *Proceedings of the IEEE* 104.2 (2016), pp. 353–373. DOI: [10.1109/JPROC.2015.2498665](https://doi.org/10.1109/JPROC.2015.2498665).
- [49] Dario Farina and Roberto Merletti. “A novel approach for precise simulation of the EMG signal detected by surface electrodes”. In: *IEEE Transactions on Bio-Medical Engineering* 48.6 (2001), pp. 637–646. DOI: [10.1109/10.923782](https://doi.org/10.1109/10.923782).
- [50] Dario Farina, Roberto Merletti, and Roger M. Enoka. “The extraction of neural strategies from the surface EMG: an update”. In: *Journal of Applied Physiology* 117.11 (2014). PMID: 25277737, pp. 1215–1230. DOI: [10.1152/japplphysiol.00162.2014](https://doi.org/10.1152/japplphysiol.00162.2014). eprint: <https://doi.org/10.1152/japplphysiol.00162.2014>. URL: <https://doi.org/10.1152/japplphysiol.00162.2014>.

- [51] Dario Farina, Francesco Negro, and Jakob L. Dideriksen. “The effective neural drive to muscles is the common synaptic input to motor neurons”. In: *The Journal of Physiology* 592.16 (Aug. 2014), pp. 3427–3441. DOI: [10.1113/jphysiol.2014.273581](https://doi.org/10.1113/jphysiol.2014.273581).
- [52] Dario Farina, Francesco Negro, and Jakob Lund Dideriksen. “The effective neural drive to muscles is the common synaptic input to motor neurons”. en. In: *J. Physiol.* 592.16 (Aug. 2014), pp. 3427–3441.
- [53] Dario Farina et al. “Detecting the Unique Representation of Motor-Unit Action Potentials in the Surface Electromyogram”. In: *Journal of Neurophysiology* 100.3 (2008). PMID: 18497352, pp. 1223–1233. DOI: [10.1152/jn.90219.2008](https://doi.org/10.1152/jn.90219.2008). eprint: <https://doi.org/10.1152/jn.90219.2008>. URL: <https://doi.org/10.1152/jn.90219.2008>.
- [54] Dario Farina et al. “Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation”. In: *Nature Biomedical Engineering* 1.2 (2017), p. 0025.
- [55] Dario Farina et al. “The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22.4 (2014), pp. 797–809. DOI: [10.1109/TNSRE.2014.2305111](https://doi.org/10.1109/TNSRE.2014.2305111).
- [56] Todd R. Farrell and Richard F. ff. Weir. “A Comparison of the Effects of Electrode Implantation and Targeting on Pattern Classification Accuracy for Prosthesis Control”. In: *IEEE Transactions on Biomedical Engineering* 55.9 (2008), pp. 2198–2211. DOI: [10.1109/TBME.2008.923917](https://doi.org/10.1109/TBME.2008.923917).
- [57] Mary Kay Floeter. “Structure and function of muscle fibers and motor units”. In: *Disorders of Voluntary Muscle*. Ed. by Andrew E. Harding and Schoenmakers, Eric I. Cambridge University Press, 2010, pp. 3–32.
- [58] Juan A. Gallego et al. “Influence of common synaptic input to motor neurons on the neural drive to muscle in essential tremor”. In: *Journal of Neurophysiology* 113.1 (2015). PMID: 25274343, pp. 182–191. DOI: [10.1152/jn.00531.2014](https://doi.org/10.1152/jn.00531.2014). eprint: <https://doi.org/10.1152/jn.00531.2014>. URL: <https://doi.org/10.1152/jn.00531.2014>.
- [59] Jonas Gehring et al. “Convolutional Sequence to Sequence Learning”. In: *CoRR* abs/1705.03122 (2017). arXiv: [1705.03122](https://arxiv.org/abs/1705.03122). URL: <http://arxiv.org/abs/1705.03122>.
- [60] Weidong Geng et al. “Gesture recognition by instantaneous surface EMG images”. In: *Scientific Reports* 6.1 (2016), p. 36571. DOI: [10.1038/srep36571](https://doi.org/10.1038/srep36571). URL: <https://doi.org/10.1038/srep36571>.
- [61] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [62] D. Halatsis et al. *A Biophysical-Model-Informed Source Separation Framework For EMG Decomposition*. 2025. arXiv: [2510.17822](https://arxiv.org/abs/2510.17822) [q-bio.NC]. URL: <https://arxiv.org/abs/2510.17822>.
- [63] L. Hargrove, K. Englehart, and B. Hudgins. “The effect of electrode displacements on pattern recognition based myoelectric control”. In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. 2006, pp. 2203–2206. DOI: [10.1109/IEMBS.2006.260681](https://doi.org/10.1109/IEMBS.2006.260681).
- [64] Levi Hargrove, Kevin Englehart, and Bernard Hudgins. “A training strategy to reduce classification degradation due to electrode displacements in pattern recognition based myoelectric control”. In: *Biomedical Signal Processing and Control* 3.2 (2008). Surface Electromyography, pp. 175–180. ISSN: 1746-8094. DOI: <https://doi.org/10.1016/j.bspc.2007.11.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1746809407001012>.
- [65] Jiayuan He et al. “A Novel Framework Based on Position Verification for Robust Myoelectric Control Against Sensor Shift”. In: *IEEE Sensors Journal* 19.21 (2019), pp. 9859–9868. DOI: [10.1109/JSEN.2019.2927325](https://doi.org/10.1109/JSEN.2019.2927325).

- [66] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: [1512.03385](https://arxiv.org/abs/1512.03385). URL: <http://arxiv.org/abs/1512.03385>.
- [67] Cj Heckman and Roger M. Enoka. “Physiology of the motor neuron and the motor unit”. English (US). In: *Handbook of Clinical Neurophysiology* 4.C (Dec. 2004), pp. 119–147. ISSN: 1567-4231. DOI: [10.1016/S1567-4231\(04\)04006-7](https://doi.org/10.1016/S1567-4231(04)04006-7).
- [68] Geoffrey Hinton et al. “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 82–97. DOI: [10.1109/MSP.2012.2205597](https://doi.org/10.1109/MSP.2012.2205597).
- [69] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [70] Ales Holobar and Damjan Zazula. “Multichannel Blind Source Separation Using Convolution Kernel Compensation”. In: *IEEE Transactions on Signal Processing* 55.9 (2007), pp. 4487–4496. DOI: [10.1109/TSP.2007.896108](https://doi.org/10.1109/TSP.2007.896108).
- [71] Aleš Holobar and Dario Farina. “Surface EMG Signal Decomposition”. In: *Surface Electromyography: Physiology, Engineering, and Applications*. IEEE Press / Wiley, 2016. Chap. 7, pp. 220–250. DOI: [10.1002/9781119082934.ch07](https://doi.org/10.1002/9781119082934.ch07).
- [72] Aleš Holobar et al. “Estimating motor unit discharge patterns from high-density surface electromyogram”. In: *Clinical Neurophysiology* 120.3 (2009), pp. 551–562. ISSN: 1388-2457. DOI: [10.1016/j.clinph.2008.10.160](https://doi.org/10.1016/j.clinph.2008.10.160).
- [73] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8). URL: <https://www.sciencedirect.com/science/article/pii/0893608089900208>.
- [74] Xiaogang Hu, William Z Rymer, and Nina L Suresh. “Motor unit pool organization examined via spike-triggered averaging of the surface electromyogram”. en. In: *J. Neurophysiol.* 110.5 (Sept. 2013), pp. 1205–1220.
- [75] He Huang *, Todd A. Kuiken, and Robert D. Lipschutz. “A Strategy for Identifying Locomotion Modes Using Surface Electromyography”. In: *IEEE Transactions on Biomedical Engineering* 56.1 (2009), pp. 65–73. DOI: [10.1109/TBME.2008.2003293](https://doi.org/10.1109/TBME.2008.2003293).
- [76] A. Hyvarinen. “Fast and robust fixed-point algorithms for independent component analysis”. In: *IEEE Transactions on Neural Networks* 10.3 (1999), pp. 626–634. DOI: [10.1109/72.761722](https://doi.org/10.1109/72.761722).
- [77] J. Ibáñez et al. “Only the Fastest Corticospinal Fibers Contribute to β Corticomuscular Coherence”. In: *Journal of Neuroscience* 41.22 (2021), pp. 4867–4879. ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.2908-20.2021](https://doi.org/10.1523/JNEUROSCI.2908-20.2021). eprint: <https://www.jneurosci.org/content/41/22/4867.full.pdf>. URL: <https://www.jneurosci.org/content/41/22/4867>.
- [78] Max Jaderberg et al. “Spatial Transformer Networks”. In: *CoRR* abs/1506.02025 (2015). arXiv: [1506.02025](https://arxiv.org/abs/1506.02025). URL: <http://arxiv.org/abs/1506.02025>.
- [79] Xinyu Jiang et al. “Open Access Dataset, Toolbox and Benchmark Processing Results of High-Density Surface Electromyogram Recordings”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 1035–1046. DOI: [10.1109/TNSRE.2021.3082551](https://doi.org/10.1109/TNSRE.2021.3082551).
- [80] Thorsten Joachims. “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”. In: *Machine Learning: ECML-98*. Ed. by Claire Nédellec and Céline Rouveirol. Vol. 1398. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1998, pp. 137–142. ISBN: 978-3-540-69781-7. DOI: [10.1007/BFb0026683](https://doi.org/10.1007/BFb0026683).
- [81] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596 (2021), pp. 583–589. DOI: [10.1038/s41586-021-03819-2](https://doi.org/10.1038/s41586-021-03819-2).

- [82] Balint K. Hodossy et al. “Leveraging high-density EMG to investigate bipolar electrode placement for gait prediction models”. In: (2023). DOI: [10.36227/techrxiv.24082206](https://doi.org/10.36227/techrxiv.24082206).
- [83] P. Kaifosh, T.R. Reardon, and CTRL-labs at Reality Labs. “A generic non-invasive neuromotor interface for human-computer interaction”. In: *Nature* 645 (2025), pp. 702–711. DOI: [10.1038/s41586-025-09255-w](https://doi.org/10.1038/s41586-025-09255-w). URL: <https://doi.org/10.1038/s41586-025-09255-w>.
- [84] Steven M. Kay. *Fundamentals of statistical signal processing: estimation theory*. USA: Prentice-Hall, Inc., 1993. ISBN: 0133457117.
- [85] István Ketykó, Ferenc Kovács, and Krisztián Zsolt Varga. “Domain Adaptation for sEMG-based Gesture Recognition with Recurrent Neural Networks”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–7. DOI: [10.1109/IJCNN.2019.8852018](https://doi.org/10.1109/IJCNN.2019.8852018).
- [86] Rami N. Khushaba and Sarath Kodagoda. “Electromyogram (EMG) feature reduction using Mutual Components Analysis for multifunction prosthetic fingers control”. In: *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*. 2012, pp. 1534–1539. DOI: [10.1109/ICARCV.2012.6485374](https://doi.org/10.1109/ICARCV.2012.6485374).
- [87] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG]. URL: <https://arxiv.org/abs/1412.6980>.
- [88] Sotiris B. Kotsiantis. “Decision trees: a recent overview”. In: *Artificial Intelligence Review* 39.4 (2013), pp. 261–283. DOI: [10.1007/s10462-011-9272-4](https://doi.org/10.1007/s10462-011-9272-4).
- [89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [90] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [91] Daniel D. Lee and H. Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791. DOI: [10.1038/44565](https://doi.org/10.1038/44565). URL: <https://doi.org/10.1038/44565>.
- [92] Jongpil Lee et al. “Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms”. In: *CoRR* abs/1703.01789 (2017). arXiv: [1703.01789](https://arxiv.org/abs/1703.01789). URL: [http://arxiv.org/abs/1703.01789](https://arxiv.org/abs/1703.01789).
- [93] Ronald S. Lefever and Carlo J. De Luca. “A Procedure for Decomposing the Myoelectric Signal Into Its Constituent Action Potentials - Part I: Technique, Theory, and Implementation”. In: *IEEE Transactions on Biomedical Engineering* BME-29.3 (1982), pp. 149–157. DOI: [10.1109/TBME.1982.324881](https://doi.org/10.1109/TBME.1982.324881).
- [94] Eric C Leuthardt et al. “A brain–computer interface using electrocorticographic signals in humans*”. In: *Journal of Neural Engineering* 1.2 (June 2004), p. 63. DOI: [10.1088/1741-2560/1/2/001](https://doi.org/10.1088/1741-2560/1/2/001). URL: <https://doi.org/10.1088/1741-2560/1/2/001>.
- [95] O. C. J. Lippold. “The relation between integrated action potentials in a human muscle and its isometric tension”. In: *The Journal of Physiology* 117.4 (1952), pp. 492–499. DOI: [10.1113/jphysiol.1952.sp004763](https://doi.org/10.1113/jphysiol.1952.sp004763). URL: <https://doi.org/10.1113/jphysiol.1952.sp004763>.
- [96] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94).
- [97] Wenjie Luo et al. “Understanding the Effective Receptive Field in Deep Convolutional Neural Networks”. In: *CoRR* abs/1701.04128 (2017). arXiv: [1701.04128](https://arxiv.org/abs/1701.04128). URL: <http://arxiv.org/abs/1701.04128>.
- [98] Ke Ma et al. “sEMG-Based Trunk Compensation Detection in Rehabilitation Training”. In: *Frontiers in Neuroscience* Volume 13 - 2019 (2019). ISSN: 1662-453X. DOI: [10.3389/fnins.2019](https://doi.org/10.3389/fnins.2019).

01250. URL: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2019.01250>.
- [99] Kostiantyn Maksymenko et al. “A myoelectric digital twin for fast and realistic modelling in deep learning”. In: *Nature Communications* 14.1 (Mar. 2023), p. 1600. DOI: [10.1038/s41467-023-37238-w](https://doi.org/10.1038/s41467-023-37238-w). URL: <https://doi.org/10.1038/s41467-023-37238-w>.
 - [100] E Martinez-Valdes et al. “Tracking motor units longitudinally across experimental sessions with high-density surface electromyography”. en. In: *J. Physiol.* 595.5 (Mar. 2017), pp. 1479–1496.
 - [101] M. D. McKay, R. J. Beckman, and W. J. Conover. “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”. In: *Technometrics* 21.2 (1979), pp. 239–245. ISSN: 00401706. URL: <http://www.jstor.org/stable/1268522> (visited on 12/21/2025).
 - [102] W. B. McKay et al. “Clinical Neurophysiological Assessment of Residual Motor Control in Post-Spinal Cord Injury Paralysis”. In: *Neurorehabilitation and Neural Repair* 18.3 (2004). PMID: 15375274, pp. 144–153. DOI: [10.1177/0888439004267674](https://doi.org/10.1177/0888439004267674). eprint: <https://doi.org/10.1177/0888439004267674>. URL: <https://doi.org/10.1177/0888439004267674>.
 - [103] I. Mendez et al. “Evaluation of the Myo armband for the classification of hand motions”. In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. 2017, pp. 1211–1214. DOI: [10.1109/ICORR.2017.8009414](https://doi.org/10.1109/ICORR.2017.8009414).
 - [104] Irene Mendez Guerra et al. “Far-field electric potentials provide access to the output from the spinal cord from wrist-mounted sensors”. In: *Journal of Neural Engineering* 19.2 (Apr. 2022), p. 026031. DOI: [10.1088/1741-2552/ac5f1a](https://doi.org/10.1088/1741-2552/ac5f1a). URL: <https://doi.org/10.1088/1741-2552/ac5f1a>.
 - [105] R. Merletti, A. Botter, and U. Barone. “Detection and Conditioning of Surface EMG Signals”. In: *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd, 2016. Chap. 3, pp. 1–37. ISBN: 9781119082934. DOI: <https://doi.org/10.1002/9781119082934.ch03>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119082934.ch03>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119082934.ch03>.
 - [106] R. Merletti, T. M. Vieira, and D. Farina. “Techniques for Information Extraction from the Surface EMG Signalhigh-Density Surface EMG”. In: *Surface Electromyography : Physiology, Engineering, and Applications*. John Wiley & Sons, Ltd, 2016. Chap. 5, pp. 126–157. ISBN: 9781119082934. DOI: <https://doi.org/10.1002/9781119082934.ch05>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119082934.ch05>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119082934.ch05>.
 - [107] Roberto Merletti and Dario Farina. “EMG Modeling and Simulation”. In: *Surface Electromyography: Physiology, Engineering, and Applications*. 2016, pp. 210–246. DOI: [10.1002/9781119082934.ch08](https://doi.org/10.1002/9781119082934.ch08).
 - [108] Roberto Merletti and Dario Farina. “Surface Electromyography for MAN-Machine Interfacing in Rehabilitation Technologies”. In: *Surface Electromyography: Physiology, Engineering, and Applications*. 2016, pp. 540–560. DOI: [10.1002/9781119082934.ch20](https://doi.org/10.1002/9781119082934.ch20).
 - [109] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: [1301.3781 \[cs.CL\]](https://arxiv.org/abs/1301.3781). URL: <https://arxiv.org/abs/1301.3781>.
 - [110] Luís Moreira et al. “Lower limb kinematic, kinetic, and EMG data from young healthy humans during walking at controlled speeds”. In: *Scientific Data* 8.1 (Apr. 2021), p. 103. ISSN: 2052-4463. DOI: [10.1038/s41597-021-00881-3](https://doi.org/10.1038/s41597-021-00881-3). URL: <https://doi.org/10.1038/s41597-021-00881-3>.
 - [111] Christoph Mulert. “Simultaneous EEG and fMRI: towards the characterization of structure and dynamics of brain networks”. In: *Dialogues in Clinical Neuroscience* 15.3 (2013). PMID: 24174908, pp. 381–386. DOI: [10.31887/DCNS.2013.15.3/cmulert](https://doi.org/10.31887/DCNS.2013.15.3/cmulert). eprint: <https://doi.org/10.31887/DCNS.2013.15.3/cmulert>.

- 31887/DCNS.2013.15.3/cmulert. URL: <https://doi.org/10.31887/DCNS.2013.15.3/cmulert>.
- [112] Vinod Nair and Geoffrey E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML’10. Haifa, Israel: Omnipress, 2010, pp. 807–814. ISBN: 9781605589077.
 - [113] Nurhazimah Nazmi et al. “A Review of Classification Techniques of EMG Signals during Isotonic and Isometric Contractions”. In: *Sensors* 16.8 (2016). ISSN: 1424-8220. DOI: [10.3390/s16081304](https://doi.org/10.3390/s16081304). URL: <https://www.mdpi.com/1424-8220/16/8/1304>.
 - [114] Francesco Negro and Dario Farina. “Linear transmission of cortical oscillations to the neural drive to muscles is mediated by common projections to populations of motoneurons in humans”. In: *The Journal of Physiology* 589.3 (2011), pp. 629–637. DOI: [10.1113/jphysiol.2010.202473](https://doi.org/10.1113/jphysiol.2010.202473). URL: <https://doi.org/10.1113/jphysiol.2010.202473>.
 - [115] Francesco Negro, Aleš Holobar, and Dario Farina. “Fluctuations in isometric muscle force can be described by one linear projection of low-frequency components of motor unit discharge rates”. In: *The Journal of Physiology* 587.24 (2009), pp. 5925–5938. DOI: <https://doi.org/10.1113/jphysiol.2009.178509>. eprint: <https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.2009.178509>. URL: <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.2009.178509>.
 - [116] Francesco Negro et al. “Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation”. en. In: *J. Neural Eng.* 13.2 (Apr. 2016), p. 026027.
 - [117] Johnny L.G. Nielsen et al. “Simultaneous and Proportional Force Estimation for Multifunction Myoelectric Prostheses Using Mirrored Bilateral Training”. In: *IEEE Transactions on Biomedical Engineering* 58.3 (2011), pp. 681–688. DOI: [10.1109/TBME.2010.2068298](https://doi.org/10.1109/TBME.2010.2068298).
 - [118] Anderson Souza Oliveira and Francesco Negro. “Neural control of matched motor units during muscle shortening and lengthening at increasing velocities”. In: *Journal of Applied Physiology* 130.6 (2021). PMID: 33955258, pp. 1798–1813. DOI: [10.1152/japplphysiol.00043.2021](https://doi.org/10.1152/japplphysiol.00043.2021). eprint: <https://doi.org/10.1152/japplphysiol.00043.2021>. URL: <https://doi.org/10.1152/japplphysiol.00043.2021>.
 - [119] Daniela Souza Oliveira et al. “A direct spinal cord–computer interface enables the control of the paralysed hand in spinal cord injury”. In: *Brain* 147.10 (Mar. 2024), pp. 3583–3595. ISSN: 0006-8950. DOI: [10.1093/brain/awae088](https://doi.org/10.1093/brain/awae088). eprint: <https://academic.oup.com/brain/article-pdf/147/10/3583/59561317/awae088.pdf>. URL: <https://doi.org/10.1093/brain/awae088>.
 - [120] Paul Otten, Jonghyun Kim, and Sang Hyuk Son. “A Framework to Automate Assessment of Upper-Limb Motor Function Impairment: A Feasibility Study”. In: *Sensors* 15.8 (2015), pp. 20097–20114. ISSN: 1424-8220. DOI: [10.3390/s150820097](https://doi.org/10.3390/s150820097). URL: <https://www.mdpi.com/1424-8220/15/8/20097>.
 - [121] Francesca Palermo et al. “Analysis of the repeatability of grasp recognition for hand robotic prosthesis control based on sEMG data”. In: *IEEE International Conference on Rehabilitation Robotics*. London, UK, 2017.
 - [122] Lin Pan et al. “Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns”. In: *Journal of NeuroEngineering and Rehabilitation* 12.1 (2015), p. 110. DOI: [10.1186/s12984-015-0102-9](https://doi.org/10.1186/s12984-015-0102-9). URL: <https://doi.org/10.1186/s12984-015-0102-9>.
 - [123] Bo Pang and Lillian Lee. “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts”. In: *CoRR* cs.CL/0409058 (2004). URL: <http://arxiv.org/abs/cs.CL/0409058>.

- [124] Ki-Hee Park and Seong-Whan Lee. “Movement intention decoding based on deep learning for multiuser myoelectric interfaces”. In: *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*. 2016, pp. 1–2. DOI: [10.1109/IWW-BCI.2016.7457459](https://doi.org/10.1109/IWW-BCI.2016.7457459).
- [125] Joao Pereira et al. *Spatial Adaptation Layer: Interpretable Domain Adaptation For Biosignal Sensor Array Applications*. 2024. arXiv: [2409.08058](https://arxiv.org/abs/2409.08058) [cs.LG]. URL: <https://arxiv.org/abs/2409.08058>.
- [126] Joao Pereira et al. “Tackling Electrode Shift in Gesture Recognition with HD-EMG Electrode Subsets”. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2024, pp. 1786–1790. DOI: [10.1109/ICASSP48485.2024.10448329](https://doi.org/10.1109/ICASSP48485.2024.10448329).
- [127] Matthew E. Peters et al. “Deep contextualized word representations”. In: *CoRR* abs/1802.05365 (2018). arXiv: [1802.05365](https://arxiv.org/abs/1802.05365). URL: <http://arxiv.org/abs/1802.05365>.
- [128] Angkoon Phinyomark et al. “EMG feature evaluation for improving myoelectric pattern recognition robustness”. In: *Expert Systems with Applications* 40.12 (2013), pp. 4832–4840. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2013.02.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417413001395>.
- [129] M. Piasecki et al. “Failure to expand the motor unit size to compensate for declining motor unit numbers distinguishes sarcopenic from non-sarcopenic older men”. In: *The Journal of Physiology* 596.9 (2018), pp. 1627–1637. DOI: <https://doi.org/10.1113/JP275520>. eprint: <https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/JP275520>. URL: <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/JP275520>.
- [130] Gianluca Piervirgili, Federico Petracca, and Roberto Merletti. “A new method to assess skin treatments for lowering the impedance and noise of individual gelled Ag–AgCl electrodes”. In: *Physiological Measurement* 35.10 (2014), pp. 2101–2112.
- [131] Stefano Pizzolato et al. “Comparison of six electromyography acquisition setups on hand movement classification tasks”. In: *PLoS One* 12.10 (2017), e0186132. DOI: [10.1371/journal.pone.0186132](https://doi.org/10.1371/journal.pone.0186132).
- [132] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: 2019. URL: <https://api.semanticscholar.org/CorpusID:160025533>.
- [133] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *CoRR* abs/1506.02640 (2015). arXiv: [1506.02640](https://arxiv.org/abs/1506.02640). URL: <http://arxiv.org/abs/1506.02640>.
- [134] H. Robbins and S. Monro. “A stochastic approximation method”. In: *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.
- [135] K. Roeleveld et al. “The motor unit potential distribution over the skin surface and its use in estimating the motor unit location”. In: *Acta Physiologica Scandinavica* 161.4 (1997), pp. 465–472. DOI: <https://doi.org/10.1046/j.1365-201X.1997.00247.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1046/j.1365-201X.1997.00247.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-201X.1997.00247.x>.
- [136] Marisol Rojas-Martínez, Miguel A. Mañanas, and Joan F. Alonso. “High-density surface EMG maps from upper-arm and forearm muscles”. In: *Journal of NeuroEngineering and Rehabilitation* 9.1 (2012), p. 85. DOI: [10.1186/1743-0003-9-85](https://doi.org/10.1186/1743-0003-9-85). URL: <https://doi.org/10.1186/1743-0003-9-85>.
- [137] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536. DOI: [10.1038/323533a0](https://doi.org/10.1038/323533a0).
- [138] Sasha Salter et al. *emg2pose: A Large and Diverse Benchmark for Surface Electromyographic Hand Pose Estimation*. 2024. arXiv: [2412.02725](https://arxiv.org/abs/2412.02725) [cs.CV]. URL: <https://arxiv.org/abs/2412.02725>.

- [139] Eric W. Sellers and Emanuel Donchin. “A P300-based brain-computer interface: initial tests by ALS patients”. In: *Clinical Neurophysiology* 117.3 (2006), pp. 538–548. DOI: [10.1016/j.clinph.2005.06.027](https://doi.org/10.1016/j.clinph.2005.06.027). URL: <https://doi.org/10.1016/j.clinph.2005.06.027>.
- [140] Jonathon W. Sensinger, Blair A. Lock, and Todd A. Kuiken. “Adaptive Pattern Recognition of Myoelectric Signals: Exploration of Conceptual Framework and Practical Algorithms”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 17.3 (2009), pp. 270–278. DOI: [10.1109/TNSRE.2009.2023282](https://doi.org/10.1109/TNSRE.2009.2023282).
- [141] Jamie Shotton et al. “Real-time human pose recognition in parts from single depth images”. In: *CVPR 2011*. 2011, pp. 1297–1304. DOI: [10.1109/CVPR.2011.5995316](https://doi.org/10.1109/CVPR.2011.5995316).
- [142] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations (ICLR)*. Conference Track Proceedings. San Diego, CA, USA, 2015.
- [143] Viswanath Sivakumar et al. *emg2qwerty: A Large Dataset with Baselines for Touch Typing using Surface Electromyography*. 2025. arXiv: [2410.20081](https://arxiv.org/abs/2410.20081) [cs.LG]. URL: <https://arxiv.org/abs/2410.20081>.
- [144] Antonietta Stango, Francesco Negro, and Dario Farina. “Spatial Correlation of High Density EMG Signals Provides Features Robust to Electrode Number and Shift in Pattern Recognition for Myocontrol”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 23.2 (2015), pp. 189–198. DOI: [10.1109/TNSRE.2014.2366752](https://doi.org/10.1109/TNSRE.2014.2366752).
- [145] Katherine M. Steele, Catherine Papazian, and Heather A. Feldner. “Muscle Activity After Stroke: Perspectives on Deploying Surface Electromyography in Acute Care”. In: *Frontiers in Neurology* 11 (Sept. 2020), p. 576757. DOI: [10.3389/fneur.2020.576757](https://doi.org/10.3389/fneur.2020.576757).
- [146] Ilya Sutskever et al. “On the importance of initialization and momentum in deep learning”. In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, June 2013, pp. 1139–1147. URL: <https://proceedings.mlr.press/v28/sutskever13.html>.
- [147] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 1998. ISBN: 978-0262193986.
- [148] Yaniv Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1701–1708. DOI: [10.1109/CVPR.2014.220](https://doi.org/10.1109/CVPR.2014.220).
- [149] Simone Tanzarella et al. “Synergistic Organization of Neural Inputs from Spinal Motor Neurons to Extrinsic and Intrinsic Hand Muscles”. In: *Journal of Neuroscience* 41.32 (2021), pp. 6878–6891. ISSN: 0270-6474. DOI: [10.1523/JNEUROSCI.0419-21.2021](https://doi.org/10.1523/JNEUROSCI.0419-21.2021). eprint: <https://www.jneurosci.org/content/41/32/6878.full.pdf>. URL: <https://www.jneurosci.org/content/41/32/6878>.
- [150] Dave Tkach, He Huang, and Todd A. Kuiken. “Study of stability of time-domain features for electromyographic pattern recognition”. In: *Journal of NeuroEngineering and Rehabilitation* 7.1 (May 2010), p. 21. ISSN: 1743-0003. DOI: [10.1186/1743-0003-7-21](https://doi.org/10.1186/1743-0003-7-21). URL: <https://doi.org/10.1186/1743-0003-7-21>.
- [151] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: [1706.03762](https://arxiv.org/abs/1706.03762). URL: [http://arxiv.org/abs/1706.03762](https://arxiv.org/abs/1706.03762).
- [152] Marina M.-C. Vidovic et al. “Improving the Robustness of Myoelectric Pattern Recognition for Upper Limb Prostheses by Covariate Shift Adaptation”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 24.9 (2016), pp. 961–970. DOI: [10.1109/TNSRE.2015.2492619](https://doi.org/10.1109/TNSRE.2015.2492619).
- [153] Thomas Weddell et al. “First-recruited motor units adopt a faster phenotype in amyotrophic lateral sclerosis”. In: *The Journal of Physiology* 599.17 (2021), pp. 4117–4130. DOI: <https://doi.org/10.1111/jphysiol.59917>.

- org/10.1113/JP281310. eprint: <https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/JP281310>. URL: <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/JP281310>.
- [154] Jin Wenguang, Li Yidong, and Shangyao Lin. “Design of a novel non-invasive wearable device for array surface electromyogram”. In: *Int. J. Inf. Electron. Eng.* 6.2 (2016). DOI: [doi:10.18178/ijiee.2016.6.2.611](https://doi.org/10.18178/ijiee.2016.6.2.611). URL: <http://www.ijiee.org/vol6/611-D2014203.pdf>.
 - [155] Le Wu et al. “Improved High-Density Myoelectric Pattern Recognition Control Against Electrode Shift Using Data Augmentation and Dilated Convolutional Neural Network”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28.12 (2020), pp. 2637–2646. DOI: [10.1109/TNSRE.2020.3030931](https://doi.org/10.1109/TNSRE.2020.3030931).
 - [156] Dezhen Xiong et al. “Deep Learning for EMG-based Human-Machine Interaction: A Review”. In: *IEEE/CAA Journal of Automatica Sinica* 8.3 (2021), pp. 512–533. DOI: [10.1109/JAS.2021.1003865](https://doi.org/10.1109/JAS.2021.1003865).
 - [157] Aaron J Young, Levi J Hargrove, and Todd A Kuiken. “The effects of electrode size and orientation on the sensitivity of myoelectric pattern recognition systems to electrode shift”. In: *IEEE transactions on biomedical engineering* 58.9 (2011), pp. 2537–2544.
 - [158] Aaron J. Young, Levi J. Hargrove, and Todd A. Kuiken. “Improving Myoelectric Pattern Recognition Robustness to Electrode Shift by Changing Interelectrode Distance and Electrode Configuration”. In: *IEEE Transactions on Biomedical Engineering* 59.3 (2012), pp. 645–652. DOI: [10.1109/TBME.2011.2177662](https://doi.org/10.1109/TBME.2011.2177662).
 - [159] Zhipeng Yu et al. “Surface EMG-Based Instantaneous Hand Gesture Recognition Using Convolutional Neural Network with the Transfer Learning Method”. In: *Sensors* 21.7 (2021). ISSN: 1424-8220. DOI: [10.3390/s21072540](https://doi.org/10.3390/s21072540). URL: <https://www.mdpi.com/1424-8220/21/7/2540>.
 - [160] Xu Zhang et al. “Adaptive Calibration of Electrode Array Shifts Enables Robust Myoelectric Control”. In: *IEEE Transactions on Biomedical Engineering* 67.7 (2020), pp. 1947–1957. DOI: [10.1109/TBME.2019.2952890](https://doi.org/10.1109/TBME.2019.2952890).

Appendix A

Appendix

A.1 Root Mean Square: A Physiologically Meaningful Feature

A.1.1 EMG Convolutional Model

As shown in [50], the average sEMG signal power can be approximated as a function of MU discharge rates and motor unit action potential (MUAP) waveforms. For this section, $x(t)$ denotes the continuous-time signal observed at a channel location (m, n) . The sEMG signal can be modelled as a convolutional mixture of sparse neural sources (more detailed in Section 2.4.2):

$$x(t) = \sum_{i=1}^N s_i(t) * h_i(t) + n(t) = \sum_{i=1}^N \sum_k h_i(t - t_{ik}) + n(t) \quad (\text{A.1})$$

where $s_i(t) = \sum_k \delta(t - t_{ik})$ is the spike train of motor unit i with firing times t_{ik} , $h_i(t)$ is the MUAP waveform, and $n(t)$ is additive noise. For isometric contractions, firing times are assumed to follow normal distributions about mean firing rates λ_i .

A.1.2 Mathematical Derivation

The instantaneous power of the sEMG signal is:

$$P(t) = x^2(t) = \left[\sum_{i=1}^N \sum_k h_i(t - t_{ik}) + n(t) \right]^2 \quad (\text{A.2})$$

Expanding this quadratic expression term by term:

$$\begin{aligned} P(t) &= \left(\sum_{i=1}^N \sum_k h_i(t - t_{ik}) \right)^2 + 2 \left(\sum_{i=1}^N \sum_k h_i(t - t_{ik}) \right) n(t) + n^2(t) \\ &= \sum_{i=1}^N \sum_k h_i^2(t - t_{ik}) + \sum_{i=1}^N \sum_{j \neq i} \sum_k \sum_\ell h_i(t - t_{ik}) h_j(t - t_{j\ell}) \\ &\quad + 2 \sum_{i=1}^N \sum_k h_i(t - t_{ik}) n(t) + n^2(t) \end{aligned} \quad (\text{A.3})$$

Taking the time average over window T , the mean square (MS) signal was obtained, corresponding to the average power:

$$\text{MS} = \frac{1}{T} \int_0^T x^2(t) dt = \langle P(t) \rangle_t \quad (\text{A.4})$$

This results in four types of terms to analyse:

$$\begin{aligned} \langle P(t) \rangle_t = & \sum_{i=1}^N \left\langle \sum_k h_i^2(t - t_{ik}) \right\rangle_t + \sum_{i=1}^N \sum_{j \neq i}^N \left\langle \sum_k \sum_{\ell} h_i(t - t_{ik}) h_j(t - t_{j\ell}) \right\rangle_t \\ & + 2 \sum_{i=1}^N \left\langle \sum_k h_i(t - t_{ik}) n(t) \right\rangle_t + \langle n^2(t) \rangle_t \end{aligned} \quad (\text{A.5})$$

Auto-correlation terms: For large T and stationary firing processes:

$$\left\langle \sum_k h_i^2(t - t_{ik}) \right\rangle_t \approx \lambda_i \int h_i^2(\tau) d\tau = \lambda_i E_i \quad (\text{A.6})$$

where E_i is the energy of MUAP i . This follows from ergodicity: each firing event contributes energy E_i to the power signal, and over long windows the average number of firings per unit time approaches the firing rate λ_i .

The cross-terms between different motor units can be analysed as $\langle h_i(t) * s_i(t) \cdot h_j(t) * s_j(t) \rangle_t$. While the MUAP waveforms $h_i(t)$ and $h_j(t)$ may have non-zero correlation due to volume conduction effects, the *spike train independence* creates random phase relationships between $s_i(t)$ and $s_j(t)$. Over sufficiently long windows, these random timing offsets cause the cross-correlation to average toward zero, even when the underlying MUAP shapes are correlated. The key requirement is that firing times t_{ik} and $t_{j\ell}$ are independent random variables. The signal-noise cross terms vanish because the noise $n(t)$ is independent of both the MUAP shapes and firing times, ensuring no systematic correlation between signal and noise components. The additive noise contributes: $\langle n^2(t) \rangle_t = \sigma_n^2$. Therefore, the MS signal becomes:

$$\text{MS} \approx \sum_{i=1}^N \lambda_i E_i + \sigma_n^2 \quad (\text{A.7})$$

A.1.3 Key Insights

The mathematical derivation shows that **MS is directly proportional to the firing rates (λ_i) of the underlying MUAPs and their energy (E_i)**. This establishes a direct link between the sEMG signal amplitude and the neural control signals driving muscle contraction, as not only firing rates increase with greater contraction levels, but also the number of active motor units and the average MUAP energy. Instead of MS, root mean square (RMS) values are used, as RMS has the same units as the original signal and are a well-established feature in sEMG classification, while still preserving its monotonic relationship with contraction levels that is useful for classification objectives.