

Friedrich-Alexander-Universität Erlangen-Nürnberg
Neuromuscular Physiology and Neural Interfacing Lab
Master's Degree in Medical Engineering

Academic Year 2024–2025

**Gait Phase Prediction and Muscle
Synergy Analysis in Lower Limb
Locomotion Using High-Density EMG
and Machine Learning**

Supervisor

Prof. Dr. Alessandro Del Vecchio

Graduate Student

Sathyaranarayanan Dhorali

Advisor

Vlad Cnejevici

Student ID

22984582

"Even if truth brings loss or suffering, it is better than a profitable falsehood."

THIRUVALLUVAR

Contents

Contents	I
List of Figures	IV
List of Tables	IX
List of Abbreviations	XI
List of Algorithms	XIII
Abstract	XIV
Zusammenfassung	XVI
1 Introduction	1
2 State of the Art	4
2.1 Hidden Markov Model-Based Gait Segmentation and Label Fusion Techniques	4
2.2 Inferring Gait Synergies from Reduced Electrode Sets	5
3 Theoretical Background	6
3.1 Muscles	6
3.2 Action Potentials and Muscle Fiber Activation	8
3.3 Resting Membrane Potential	10
3.4 Motor Unit Physiology and EMG Signal Generation	11
3.4.1 Gait and Muscle Synergies	13
3.4.2 Muscle Synergies with Non-Negative Matrix Factorization	14
3.4.3 Phases of the Human Gait Cycle and Joint Kinetics	15
3.5 Deep Learning	18
4 Methods	20
4.1 Data Acquisition	20
4.1.1 Marker Data Processing	22
4.1.2 Electromyography Data Alignment and Preprocessing	23
4.1.3 Feature Extraction from Surface Electromyography Signals	23
4.1.4 Muscle Synergy Extraction via Non-negative Matrix Factorization	26
4.2 Kinematic Feature Extraction	32
4.2.1 Differentiation: Linear Velocity and Acceleration	32
4.2.2 Joint Flexion Angles via Dot Product	33
4.2.3 Angular Momentum	33
4.2.4 Filtering and Noise Reduction	34

4.2.5	Windowing and Statistical Feature Computation	35
4.3	Integration with Electromyography Muscle Synergies and Model Fusion .	35
4.4	Deep Learning Models for Synergy-Based Gait Phase Prediction	38
4.4.1	Long Short-Term Memory (LSTM) Networks	38
4.4.2	Convolutional Neural Network (CNN)	44
4.4.3	Transformer Network	47
4.5	Output and Ensemble Fusion	52
4.6	Hyperparameter Tuning	53
4.6.1	Hyperparameter Tuning Objective and Cross-Validation Integration	54
4.6.2	Search Strategies for Hyperparameter Tuning	54
4.6.3	Hyperparameter Search Spaces for Different Algorithms	55
4.6.4	Integration with Cross-Validation	57
4.6.5	Train–Test Split and Final Evaluation	58
4.7	Final Model Selection and Retraining	58
4.8	Training Configuration and Optimization	59
4.8.1	Loss Function	60
5	Results	61
5.1	Synergy–Phase Alignment with Gait Subphases	61
5.1.1	Statistical Comparison of Synergy Activation Across Walking Conditions	62
5.2	Statistical Analysis of Muscle Synergy Patterns	66
5.2.1	Friedman Test Analysis	66
5.3	Statistical Analysis of Reconstruction Methods	68
5.3.1	Friedman Test for Reconstruction Methods	69
5.3.2	Nemenyi Post-Hoc Test	70
5.3.3	Mann–Whitney U Tests	70
5.4	Evaluation of Vastus Lateralis→Tibialis Anterior Reconstruction	73
5.4.1	Static Vastus Lateralis → Tibialis Anterior Correlation Analysis .	74
5.4.2	Convolutional Neural Network–Bidirectional Long Short-Term Memory Temporal Refinement	74
5.4.3	Walking Mode Comparison of Convolutional Neural Network–Bidirectional Long Short-Term Memory Performance	77
5.4.4	Comparison of Static versus Convolutional Neural Network–Bidirectional Long Short-Term Memory Reconstruction	78
6	Discussion	79
6.1	Synergy–Phase Alignment Across Walking Modes	79
6.2	Five Muscle Synergies Are Required to Reconstruct Steady-State Electromyographic Signals	79

6.3	Reconstruction Methods: Statistical Equivalence with a Trend Toward Anatomical Specificity	80
6.4	Temporal Convolutional Neural Network–Bidirectional Long Short-Term Memory Model Greatly Improves Vastus Lateralis to Tibialis Anterior Reconstruction	80
6.5	Physiological and Clinical Implications	80
6.6	Study Limitations	81
7	Conclusion	82
8	Appendix	84
	Bibliography	89
	Acknowledgements	99
	Declaration of authorship	100

List of Figures

Figure 3.1:	A. Skeletal muscle—long, striated fibres with nuclei at the fibre edges. B. Cardiac muscle—branched, striated fibres with central nuclei and intercalated disks. C. Smooth muscle—spindle-shaped, non-striated fibres with centrally located nuclei. Adapted from López [18].	7
Figure 3.2:	Structure of a single sarcomere, the fundamental unit of a muscle fiber. The Z-lines define the boundaries of the sarcomere and anchor the thin actin filaments. The light I-band contains only actin, while the darker A-band shows the region where actin and myosin filaments overlap. At the center of the A-band lies the H-zone, made up of myosin only, and the M-line, which helps stabilize the thick filaments during contraction. Adapted from López [18].	8
Figure 3.3:	Neurophysiology of a motor unit: A neural signal travels from the brain to the spinal cord, reaching an α -motor neuron. The neuron projects to the muscle's innervation zone, where it branches and activates a specific group of muscle fibers. This motor neuron and its fibers form the motor unit. Adapted from Cnejevici [22].	9
Figure 3.4:	Time course of a muscle fibre action potential. At rest (1), the sarcolemma maintains a membrane potential of approximately -90 mV. Upon stimulation (2), increased sodium (Na^+) permeability causes rapid depolarization. The membrane potential peaks around $+10$ mV (3), followed by repolarization (4) during the refractory period, as potassium (K^+) exits the cell. Finally, the sodium-potassium pump restores the original ion gradients (5), returning the membrane to its resting potential. Adapted from Kamen [23].	11
Figure 3.5:	Motor unit recruitment and firing rate modulation. Smaller motor units are recruited first, producing slower and lower-force muscle twitches. As muscle demand increases, larger motor units are progressively recruited, generating stronger and faster twitches. Total force output rises through two mechanisms: the sequential recruitment of motor units and the increase in their individual discharge rates, as shown in the lower graph. Adapted from Cnejevici [22].	12

Figure 3.6: Creation of the surface EMG signal. Each motor unit pulse train (drive to motor units) is convolved with its corresponding motor unit action potential (MUAP) waveform. The resulting MUAP pulse trains from multiple motor units are summed together—along with background noise—to form the composite surface EMG signal. This reflects the total electrical activity recorded during muscle activation. Adapted from Cnejevici [22].

13

Figure 3.7: Visual representation of the human gait cycle with corresponding limb positions, support phases (double vs. single), and stride sub-phases. Each segment is aligned with the percentage of gait cycle progression and identifies key events such as heel strike and toe-off. Adapted from Kopelovich [31].

15

Figure 4.1: (A) High-density surface EMG (HD-sEMG) signals are recorded from lower-limb muscles and data are wirelessly transmitted for real-time monitoring and storage. (B) The EMG signals are processed using a 100 ms sliding window with 99% overlap to extract temporal features and compute muscle synergies via non-negative matrix factorization (NMF); the resulting synergy activation coefficients are then used to reconstruct the original EMG signals, enhancing interpretability. (C) A fusion Hidden Markov Model (HMM), trained on both marker-derived kinematic features and EMG-based synergy activations, is used to infer gait phase labels for each frame within the stride. producing robust temporal segmentation using muscle synergies. (D) A deep learning model-selected as the best performer among BiLSTM, CNN, and Transformer architectures—is trained on the synergy features to classify gait stages with high accuracy. supporting real-time phase decoding for neuroprosthetic control.

30

Figure 4.2: Gait phase classification pipeline. The left branch processes EMG signals through feature extraction, synergy analysis, and reduced-channel reconstruction, enabling CNN–BiLSTM-based prediction of tibialis anterior (TA) from vastus lateralis (VL). The right branch handles marker-based kinematics, extracting velocity and momentum features. Both branches yield statistical features used to train HMMs for gait segmentation. Outputs are fused and classified into gait phases: Loading Response (LR), Mid Stance (MS), Terminal Stance (TS), Pre Swing (PS), and Terminal Swing (TS). Adapted from Cnejevici [22].

31

Figure 4.3: Five-state left-to-right (Bakis) Hidden Markov Model with cyclic return. Each state S_i corresponds to a discrete gait phase, with self-loop transitions (green) representing within-phase persistence ($\alpha, \beta, \gamma, \delta, \theta$) and forward transitions (orange) denoting progression through the gait cycle. A cyclic transition (red) from S_5 back to S_1 enables continuous modeling of successive gait strides. Adapted from Roth [13].

37

Figure 4.4: LSTM architecture: The Long Short-Term Memory (LSTM) cell consists of three main gates—forget gate, input gate, and output gate—that regulate the flow of information. The forget gate (f_t) controls which parts of the previous cell state (C_{t-1}) are retained. The input gate (i_t) and candidate state (\tilde{C}_t) update the cell state (C_t). The output gate (o_t) determines which parts of the cell state contribute to the output hidden state (h_t). This gated mechanism allows the LSTM to maintain long-term dependencies while mitigating vanishing gradients during training. Adapted from Zhan et al. [59].

39

Figure 4.5: Attention layer architecture: The input sequence is first processed by a bidirectional LSTM (BiLSTM), generating forward and backward hidden states that are concatenated and linearly projected into query (Q), key (K), and value (V) vectors—where Q acts as a “question” at each time step, K as an “index” to compare against, and V as the “content” to be aggregated—before being passed to a self-attention layer. Scaled dot-product attention then computes attention weights that highlight the most relevant time steps, and the resulting weighted sum forms a global context vector z that captures the most informative temporal features, which is finally used for classification. Adapted from Zhan et al. [59].

41

Figure 4.6: Overview of the Transformer-based time-series classification model. The raw input sequence is first embedded via a 1D convolutional layer and enriched with learned positional encodings. The resulting sequence is processed by N pre-norm Transformer encoder blocks—each consisting of multi-head self-attention, a two-layer feed-forward network with ReLU and dropout, residual connections, and layer normalization. The final hidden states are aggregated (mean, max, or first-step pooling) into a fixed-length context vector, which is passed through a ReLU-activated dense layer with dropout and a linear classifier to produce logits for softmax or sigmoid probability estimation. Adapted from Carneros-Prado et al. [69]

48

Figure 5.1: Contour plot of synergy activation over the gait cycle for Overground walking. Synergy 1 peaks at 80%–100% (Terminal Swing), Synergy 2 peaks at 0%–10% (Loading Response), Synergy 3 peaks at 50%–60% (Terminal Stance), Synergy 4 peaks at 60%–80% (Pre-Swing), and Synergy 5 peaks at 10%–50% (Mid-Stance).	64
Figure 5.2: Contour plot of synergy activation over the gait cycle for Treadmill walking. Synergy 1 peaks at 0%–10% (Loading Response), Synergy 2 peaks at 50%–60% (Terminal Stance), Synergy 3 peaks at 60%–80% (Pre-Swing), Synergy 4 peaks at 10%–50% (Mid-Stance), and Synergy 5 peaks at 80%–100% (Terminal Swing).	65
Figure 5.3: Participant-wise variance explained as a function of synergy count. Each line represents one participant’s variance reconstruction performance for 1 to 5 extracted synergies. The increasing trend supports the selection of higher synergy counts, aligning with the Friedman test results.	67
Figure 5.4: Plateau focus on explained variance for 4 vs. 5 synergies. Bars denote the group mean \pm SD, while individual colored lines trace each participant’s increase.	67
Figure 5.5: Plateau focus on explained variance for 3 vs. 4 synergies. Bars denote the group mean \pm SD, while individual colored lines trace each participant’s increase.	67
Figure 5.6: Explained-variance scores by participant and reconstruction method.	69
Figure 5.7: Violin distributions of explained-variance scores for each reconstruction method, comparing Overground (left half of each violin) versus Treadmill (right half of each violin) walking.	71
Figure 5.8: Bivariate joint-plots of synergy counts versus explained-variance score for each participant, shown separately for Overground (left) and Treadmill (right) walking.	72
Figure 5.9: Radar plots of mean TA reconstruction correlation (\pm SD) for Overground (left) and Treadmill (right) walking.	75
Figure 5.10: CNN–BiLSTM–reconstructed TA signal (dashed orange) overlaid on the original TA signal (solid blue) for a representative participant. .	76
Figure 5.11: CNN–BiLSTM–reconstructed TA signal (dashed orange) overlaid on the original TA signal (solid blue) for a representative participant. .	76
Figure 8.1: Summary of model-wise participant evaluation. Subfigures (a–d) display radar plots of the combined score—calculated as the mean of weighted F1 score and weighted ROC AUC—for each participant (P001–P007) across four architectures: CNN, BiLSTM, Transformer, and the Ensemble model.	86

Figure 8.2: Distribution of combined validation scores, defined as the mean of the weighted F1 score and weighted ROC AUC (Equation 4.52), for three model architectures—BiLSTM, CNN, and Transformer—across seven participants. Each model was optimized via Bayesian Optimization, Random Search, and Hyperband. The Transformer consistently achieved the highest average performance and the lowest inter-run variability.	87
Figure 8.3: Participant-wise variability in gait cycle durations, shown as boxplots with swarm overlays. Despite individual differences, the majority of cycles lie between 1.5s and 2.0s.	87
Figure 8.4: Phase-wise duration distributions for all detected gait phases, with swarm points indicating individual stride events. Boxplots correspond to: LR = Loading Response, MS = Midstance, TS = Terminal Stance, PS = Preswing, and TS = Terminal Swing.	88

List of Tables

Table 1:	Range of motion (ROM) requirements at the hip, knee, and ankle joints during each sub-phase of the gait cycle.	15
Table 2:	Participant allocation across walking conditions and corresponding high-density EMG probe configurations.	22
Table 3:	Layer-by-layer breakdown of the proposed BiLSTM-based classification model. The architecture includes optional stacked BiLSTM layers, self-attention with mean pooling, and a feedforward classification head. Tensor dimensions are defined using symbols listed in Table 4.	43
Table 4:	Definition of tensor shape symbols used throughout the BiLSTM model description in Table 3.	43
Table 5:	Layer-by-layer architecture of the CNN-based classification model. The network processes raw sequential input using stacked 1D convolutional layers with optional dropout and non-linear activation, followed by global max pooling and fully connected layers. The final output layer produces unnormalized class logits. For symbol definitions and hyperparameter explanations, see Table 6.	46
Table 6:	Definitions of symbols, shapes, and hyperparameters used throughout the CNN-based model description in Table 5.	46
Table 7:	Layer-by-layer architecture of the Transformer-based time-series classification model. The network embeds the input sequence using a 1D convolutional layer, augments it with learned positional encodings, and processes it through N stacked Transformer blocks. Each block includes multi-head self-attention, a feedforward network, residual connections, and layer normalization. Global pooling is used to generate a fixed-length representation, followed by a dense classification head. Symbol definitions are listed in Table 8.	51
Table 8:	Definitions of tensor dimensions and hyperparameters used in the Transformer-based model (Table 7).	52
Table 9:	Hyperparameter search space for the LSTM-based model. Parameters include the number of LSTM units, bidirectionality, optional second LSTM layers, dropout rates, fully connected layer size, and learning rate.	56

Table 10: Hyperparameter search space for the CNN-based model. This includes the number of convolutional layers, filters, kernel sizes, activation functions, dropout configuration, and the size of the final dense layer.	56
Table 11: Hyperparameter search space for the Transformer-based model. Search dimensions include embedding size, the number of encoder blocks and attention heads, feed-forward network sizes, dropout rates, and final dense layer configuration.	57
Table 12: Participant-level Mann–Whitney U results for each synergy–phase pair Overground=4, Treadmill=3. No comparison survives Benjamini–Hochberg FDR correction ($q = 0.05$).	63
Table 13: Dominant muscle synergy–gait phase mapping for overground and treadmill walking.	63
Table 14: Post-hoc Nemenyi pairwise comparisons of mean rank for explained variance across reconstruction methods.	70
Table 15: Mann–Whitney U test results comparing explained variance between overground and treadmill walking for each reconstruction method. . . .	71
Table 16: Participant-wise mean \pm SD of Pearson r for VL \rightarrow TA reconstruction by the Static Method	74
Table 17: Participant-wise mean \pm SD of Pearson r for VL \rightarrow TA reconstruction by the CNN–BiLSTM model.	75
Table 18: Best hyperparameter configuration of the BiLSTM-based model, selected across all participants. The configuration includes hidden size, dropout settings, and optimizer parameters. Optional second LSTM layer was disabled during final training.	84
Table 19: Best hyperparameter configuration of the CNN-based model, optimized across all participants. This includes convolutional depth, kernel parameters, dropout rates, and fully connected layer settings.	85
Table 20: Best hyperparameter configuration of the Transformer-based model, determined across all participants. Includes embedding dimensions, number of attention heads, feed-forward network size, dropout rates, and optimizer learning rate.	85

List of Abbreviations

1D CNN	One-Dimensional Convolutional Neural Network
AAC	Average Amplitude Change
A/D	Analog-to-Digital Converter
ANOVA	Analysis of Variance
ASHA	Asynchronous Successive Halving Algorithm
ATP	Adenosine Triphosphate
BiLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CPG	Central Pattern Generator
DTW	Dynamic Time Warping
ECG	Electrocardiography
EMG	Electromyography
FFN	Feed-Forward Network
GRF	Ground Reaction Force
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
ICA	Independent Component Analysis
IAV	Integrated Absolute Value
IC	Initial Contact
K-fold	K-Fold Cross-Validation
L1 / L2	ℓ_1 and ℓ_2 Regularization
LSTM	Long Short-Term Memory
MAV	Mean Absolute Value
MFAP	Muscle Fiber Action Potential
MU	Motor Unit
MUAP	Motor Unit Action Potential
NMF	Non-Negative Matrix Factorization
PCA	Principal Component Analysis

ReLU	Rectified Linear Unit
RMS	Root Mean Square
ROC AUC	Receiver Operating Characteristic Area Under the Curve
ROM	Range of Motion
sEMG	Surface Electromyography
SMOTE	Synthetic Minority Over-sampling Technique
TO	Toe-Off
VL	Vastus Lateralis
WL	Waveform Length

List of Algorithms

Independent Component Analysis (ICA)

Non-Negative Matrix Factorisation (NMF)

Dynamic Time Warping (DTW)

Principal Component Analysis (PCA)

Gaussian Hidden Markov Model (HMM)

Bidirectional Long Short-Term Memory (BiLSTM)

Convolutional Neural Network (1D-CNN)

Transformer

Random Search

Bayesian Optimisation

HyperBand / ASHA (Successive-Halving)

Abstract

Human walking emerges from the coordinated activation of many muscles. We recorded high-density EMG (128 channels) from seven adults during overground and treadmill walking to address three questions: (1) How many muscle synergies suffice to reconstruct EMG? (2) Are those synergies phase-locked across modes? (3) Can this enable drastic electrode reduction for wearable systems?

The preceding analyses confirmed that each muscle synergy exhibited a tightly phase-locked activation pattern, but that the dominant synergy–phase assignments were systematically re-scheduled between overground and treadmill walking. Crucially, after Benjamini–Hochberg False Discovery Rate correction at $q = 0.05$, none of the 25 synergy–phase comparisons survived FDR correction (all adjusted $p > 0.05$), indicating that amplitude distributions do not differ between conditions. Timing remains sharply phase-locked in both modes, but which synergy occupies each subphase depends on walking context.

As reported above, non-negative matrix factorization showed that increasing from three to four to five synergies raised the variance explained from $61.6\% \pm 7.3\%$ to $70.6\% \pm 6.3\%$ and $92.3\% \pm 1.2\%$, respectively. We therefore turned to the issue of dimensionality: quantifying how much additional variance in the electromyographic recordings is captured as the model complexity increases from three to five synergies.

We fused kinematic and synergy cues using two parallel Hidden Markov Models (HMMs). We then trained three sequence classifiers—a one-dimensional convolutional neural network (1D CNN), a bidirectional long short-term memory (BiLSTM) network with self-attention, and a Transformer model—on the synergy time series. The Transformer achieved weighted- $F_1 = 0.94$ and weighted ROC–AUC = 0.99, with an ensemble of all three models reaching 0.995 ± 0.004 .

To test electrode reduction, we selected vastus lateralis channels using three methods and reconstructed full EMG using a shallow neural network trained to map the reduced input to synergy activations, which were then recombined with the NMF weight matrix. The explained variance for each approach was $72\% \pm 9.76\%$ for anatomical selection (4 channels), $65.15\% \pm 10.23\%$ for variance-based selection (10 channels), and $64.12\% \pm 15.10\%$ for

ℓ_1 -based selection (10 channels). A Friedman test confirmed overall differences ($p < 0.05$), though no pairwise differences were statistically significant based on Nemenyi post-hoc comparisons.

Finally, we trained a CNN–BiLSTM model to predict tibialis anterior activity from reconstructed vastus lateralis EMG. This model achieved $r = 0.947 \pm 0.060$ (overground) and $r = 0.958 \pm 0.035$ (treadmill; $p = 0.857$), significantly outperforming a static linear mapping ($r < 0.05$; $p < 0.01$). In summary, five phase-locked synergies robustly represent gait, enable near-perfect phase recognition, and support accurate reconstruction of full EMG and muscle-specific signals from a reduced set of electrodes—advancing practical wearable EMG technologies.

Zusammenfassung

Das menschliche Gehen entsteht aus der koordinierten Aktivierung vieler Muskeln. Wir haben hochauflösendes EMG (128 Kanäle) von sieben Erwachsenen während des Gehens über Boden und auf dem Laufband aufgezeichnet, um drei Fragen zu beantworten: (1) Wie viele Muskelsynergien genügen, um EMG zu rekonstruieren? (2) Sind diese Synergien phasenstarr zwischen den Modi gekoppelt? (3) Ermöglicht dies eine drastische Elektrodenreduktion für tragbare Systeme?

Die vorangegangenen Analysen bestätigten, dass jede Muskelsynergie ein streng phasenstarr gekoppeltes Aktivierungsmuster aufweist, dass jedoch die Zuordnung der dominanten Synergie zu den Gangphasen zwischen Übergrund- und Laufbandgehen systematisch neu geplant wird. Nach Benjamini–Hochberg-Korrektur der False-Discovery-Rate bei $q = 0,05$ überstand keine der 25 Synergie-Phasen-Vergleiche die FDR-Korrektur (alle adjustierten $p > 0,05$), was darauf hinweist, dass sich die Amplitudenverteilungen zwischen den Bedingungen nicht unterscheiden. Die Zeitstruktur bleibt in beiden Modi stark phasenstarr, aber welche Synergie jede Unterphase belegt, hängt vom Gehkontext ab.

Wie oben berichtet, zeigte die nichtnegative Matrixfaktorisierung, dass eine Erhöhung von drei auf vier bzw. fünf Synergien den erklärten Varianzanteil von $61,6\% \pm 7,3\%$ auf $70,6\% \pm 6,3\%$ bzw. $92,3\% \pm 1,2\%$ steigerte. Daher widmeten wir uns der Frage der Dimensionalität: der Quantifizierung, wie viel zusätzliche Varianz in den EMG-Aufzeichnungen erfasst wird, wenn die Modellkomplexität von drei auf fünf Synergien erhöht wird.

Wir kombinierten kinematische und Synergie-Hinweise mithilfe zweier paralleler Hidden-Markov-Modelle (HMMs). Anschließend trainierten wir drei Sequenzklassifikatoren – ein eindimensionales Convolutional Neural Network (1D-CNN), ein bidirektionales Long Short-Term Memory-Netzwerk (BiLSTM) mit Self-Attention und ein Transformer-Modell – auf den Synergie-Zeitreihen. Der Transformer erreichte einen gewichteten $F_1 = 0,94$ und eine gewichtete ROC-AUC von 0,99; ein Ensemble aller drei Modelle erzielte $0,995 \pm 0,004$.

Zur Prüfung der Elektrodenreduktion wählten wir Vastus-lateralis-Kanäle mithilfe von drei Methoden aus und rekonstruierten das vollständige EMG mit einem flachen neuronalen Netz, das darauf trainiert wurde, die reduzierte Eingabe auf Synergie-Aktivierungen

abzubilden, die anschließend mit der NMF-Gewichtsmatrix rekombiniert wurden. Der erklärte Varianzanteil betrug $72\% \pm 9,76\%$ für die anatomische Auswahl (4 Kanäle), $65,15\% \pm 10,23\%$ für die varianzbasierte Auswahl (10 Kanäle) und $64,12\% \pm 15,10\%$ für die ℓ_1 -basierte Auswahl (10 Kanäle). Ein Friedman-Test bestätigte Gesamtdifferenzen ($p < 0,05$), obwohl keine paarweisen Unterschiede anhand der Nemenyi-Post-hoc-Vergleiche statistisch signifikant waren.

Schließlich trainierten wir ein CNN-BiLSTM-Modell, um die Aktivität des Musculus tibialis anterior aus den rekonstruierten Vastus-lateralis-EMG-Signalen vorherzusagen. Dieses Modell erzielte $r = 0,947 \pm 0,060$ (Übergrund) und $r = 0,958 \pm 0,035$ (Laufband; $p = 0,857$) und übertraf damit signifikant eine statische lineare Abbildung ($r < 0,05$; $p < 0,01$). Zusammenfassend repräsentieren fünf phasenstarre Synergien den Gang robust, ermöglichen eine nahezu perfekte Phasenerkennung und unterstützen die präzise Rekonstruktion vollständiger EMG- und muskelspezifischer Signale aus einer reduzierten Elektrodenanzahl – ein Fortschritt für praktische tragbare EMG-Technologien.

1 Introduction

Walking is an essential element of human independence—yet, despite its everyday familiarity, even subtle changes in our stride can reveal injury, disease, or the effects of training. Understanding how and why we walk differently under varying conditions is the goal of gait analysis, a field that intersects clinical rehabilitation, biomechanics research, and the design of assistive and wearable robotic devices. Today’s gait laboratories combine optical motion-capture systems to track limb movements in three dimensions, force plates to register ground reaction forces, and high-density surface electromyography (HD-sEMG) to record the electrical signals that drive muscle contractions. Together, these tools help clinicians evaluate recovery after stroke or limb loss, engineers optimize prosthetic limbs and exoskeletons, and sports scientists fine-tune athletic performance.

At the foundation of these analyses lies a key concept: the gait cycle, the fundamental unit used to describe and compare walking patterns. The gait cycle refers to the interval between two successive contacts of the same foot with the ground, typically beginning and ending with heel strike—the moment the heel first makes contact with the surface [1]. Within this cycle, human locomotion is classically divided into two main phases: the stance phase, which accounts for approximately 60 % of the cycle, and the swing phase, comprising the remaining 40 %. The stance phase begins with initial contact (IC)—commonly represented by heel strike—and ends with toe-off (TO), the moment the foot leaves the ground. The swing phase follows, spanning from TO to the next IC [2]. Precise identification of these phases is crucial for the effective control of assistive technologies, such as prosthetic limbs and exoskeletons [3].

Muscle activity during human locomotion is regulated by the central nervous system (CNS) in a coordinated manner. Rather than activating each muscle individually, the CNS controls groups of muscles through common activation patterns, commonly referred to as muscle synergies [4–7]. During locomotion, specific muscle synergies are consistently recruited at particular time points within the gait cycle [4, 8, 9]. The activation of a small number of such synergies can reproduce the complex muscle activation patterns observed during walking, effectively reducing the dimensionality of motor control [10]. For instance, Clark et al. [8] identified four basic muscle synergy factors in the healthy adult gait cycle,

whereas Ivanenko et al. [11] identified five. These synergies correspond to phase-specific functional demands, including propulsion, weight acceptance, trunk stabilization, swing, and fine-tuning [11]. Muscle synergy analysis thus offers valuable insight into the neural simplification of multi-muscle coordination and reveals how this modular organization is disrupted in motor pathologies such as stroke, where a reduced number of synergies has been shown to correlate with greater motor impairment [12].

Despite the advantages of analyzing muscle synergies, several important questions remain unresolved. This thesis aims to address some of these open questions through systematic investigation.

We employ surface electromyography (EMG) to non-invasively measure muscle activity using skin-mounted electrodes. Specifically, this study utilizes high-density EMG (HD-EMG), which involves arrays of closely spaced electrodes to capture detailed spatial and temporal activation patterns from lower-limb muscles during gait.

First, we investigate how many distinct EMG synergies are required to accurately represent gait across different individuals. Our goal is to determine the optimal number of synergies needed to represent gait in our dataset and to assess whether this number remains consistent across participants.

Second, we examine how these synergy activation patterns correspond to specific gait phases. In other words, do the extracted synergies align with key gait events such as initial contact, mid-stance, toe-off, initial swing, and terminal swing?

Third, we explore whether synergy activations can be inferred from a reduced set of EMG channels. In practical applications—such as wearable systems, mobile gait analysis setups, or clinical environments—it is often not feasible to record from all muscles involved in walking. Therefore, we evaluate whether the activation of the identified synergies can be reliably predicted from a smaller subset of electrodes.

To address these questions, this thesis proposes a hybrid approach that combines the following three components:

1. **Synergy Extraction from EMG Recordings:** We employ non-negative matrix factorization (NMF) to extract a low-dimensional set of muscle synergies from lower-limb EMG recordings, which capture the electrical activity generated by muscles during movement.

2. **Probabilistic Sequence Modeling (Hidden Markov Model, HMM):** The extracted synergy activations, along with kinematic markers, serve as observations in a hidden Markov model (HMM). This model imposes a cyclic structure on the sequential nature of gait (e.g., stance → swing → stance) and provides probabilistic labels for gait phase transitions and classification tasks.
3. **Deep Learning Classifiers:** Advanced neural networks such as bidirectional long short-term memory (BiLSTM), convolutional neural networks (CNN), and transformer models are employed to refine gait-phase classification by learning complex temporal dependencies from the input features.

2 State of the Art

2.1 Hidden Markov Model-Based Gait Segmentation and Label Fusion Techniques

Hidden Markov models (HMMs) are powerful tools for analyzing time-series or sequential data. They are particularly effective for identifying gait phases by modeling the underlying sequence of hidden states (i.e., gait phases) and associating them with observed features extracted from electromyography (EMG) signals. HMMs offer several advantages, including the ability to capture the cyclical nature of gait, model probabilistic transitions between phases, and maintain robustness in the presence of sensor noise.

1. **Marker-based HMM:** Utilizes motion capture data to model gait phase transitions [13].
2. **Synergy-based HMM:** Leverages EMG-derived muscle synergies for phase inference [13].

These two sources provide complementary information—marker data capture external kinematic movements, while muscle synergies reflect internal neuromuscular coordination. Since each method may introduce occasional errors, their combination enhances both robustness and accuracy. To the best of our knowledge, this combined approach—marker-HMM + synergy-HMM—for gait phase labeling has not been previously reported in the literature. This thesis addresses that gap by proposing a label fusion framework, where the fused labels serve as training targets for deep learning models that classify gait phases using EMG data alone.

2.2 Inferring Gait Synergies from Reduced Electrode Sets

Surface EMG electrodes are limited in their ability to record signals from deeper muscle groups, which often play a crucial role during gait. This limitation—discussed by Bogey and Ao et al. [14, 15]—restricts the completeness of muscle activation analysis, particularly in gait-related studies. This raises an important question: can the missing muscle activations be reconstructed from available EMG signals using synergy-based methods?

Ao et al. [15] introduced the concept of **Synergy Extrapolation (SynX)**. In their method, synergies were first extracted from a complete set of 16 EMG channels per leg. One EMG channel was then artificially removed, and its activation was predicted using the remaining channels. Their findings demonstrated that approximately 5–6 synergies were sufficient to reconstruct the missing muscle signals with a reasonable degree of accuracy.

Inspired by this, our method follows a similar framework and consists of the following steps:

1. **Decompose the EMG signals:** Use non-negative matrix factorization (NMF) to extract low-dimensional muscle synergies.
2. **Select channels from the vastus lateralis (VL):** Use three criteria:
 - Anatomical selection (predefined VL electrode indices)
 - Highest variance within the VL electrode block
 - Largest L1-regularized weights within the VL block
3. **Train a neural network:** Map the selected EMG channels to the corresponding synergy activations.
4. **Reconstruct full EMG signals:** Combine the network-predicted activations with the original synergy weight matrix.
5. **Evaluate reconstruction performance:** Use explained variance.
6. **Predict TA activity:** Train a CNN–BiLSTM model on the selected VL channels to predict tibialis anterior (TA) activity.
7. **Assess reconstruction quality:** Compute per-channel Pearson correlations and overlay signal traces to evaluate VL→TA prediction.

3 Theoretical Background

3.1 Muscles

A muscle is a body tissue composed of elongated cells that contract in response to stimulation, thereby producing motion. Muscles are categorized into three main types:

- **Smooth Muscle:** Smooth muscle is an involuntary type of muscle controlled by the autonomic nervous system. Unlike other muscle types, it lacks striations and contracts slowly while consuming minimal energy. It is typically found in the walls of internal organs such as the stomach and bladder. Compared to skeletal muscles, its contractions are slower and more sustained [16–18].
- **Cardiac Muscle:** Cardiac muscle is found exclusively in the heart. Although it shares structural features with skeletal muscle, it functions involuntarily—that is, without conscious control. It generates the electrical impulses recorded during electrocardiography (ECG) [16, 18].
- **Skeletal Muscle:** Skeletal muscles are the only type of muscle under voluntary control. They are essential for body movement and posture, and electromyography (EMG) signals are recorded from these muscles. Skeletal muscles attach to bones via tendons and are composed of long fibers wrapped in connective and fatty tissue. In adults, muscle fibers can reach up to 30 cm in length and approximately 100 micrometers in diameter. As shown in Figure 3.1, skeletal muscle fibers exhibit a characteristic striped (striated) appearance, with alternating light and dark bands that shift during muscle contraction and relaxation [16–18].

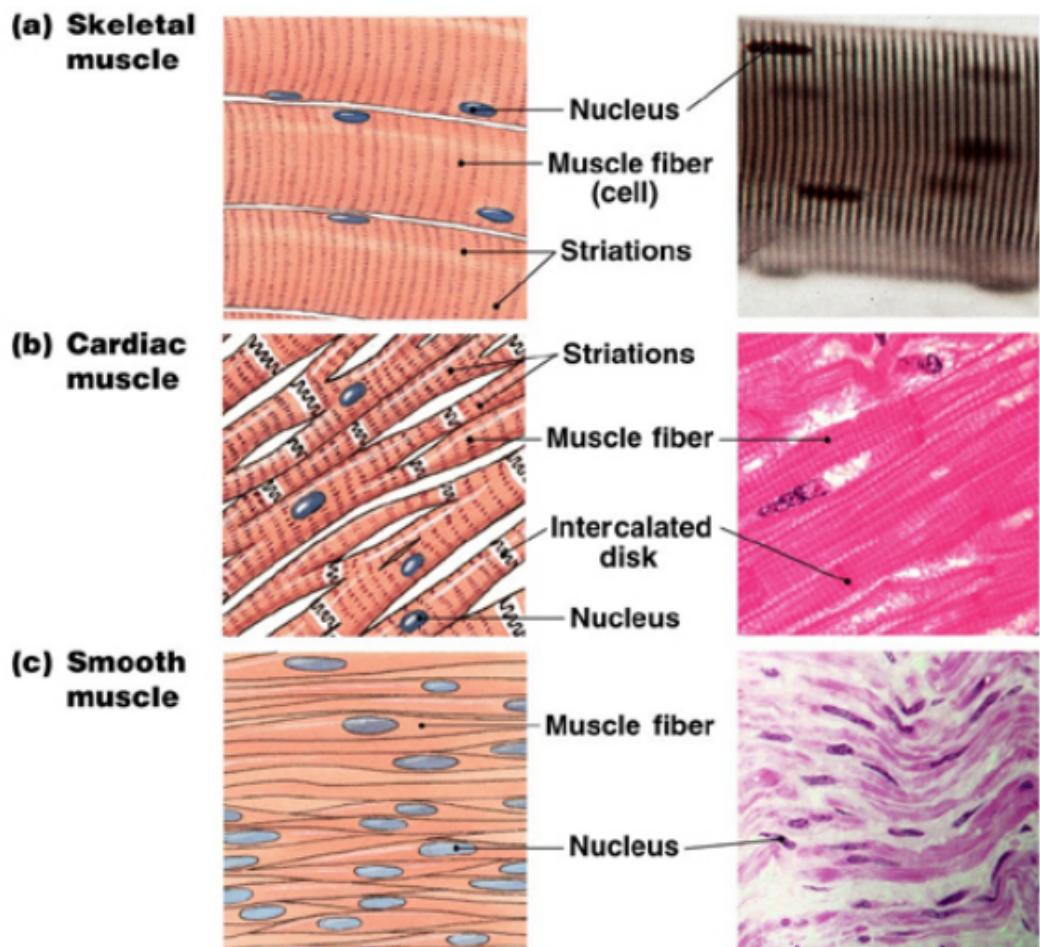


Figure 3.1: A. Skeletal muscle—long, striated fibres with nuclei at the fibre edges. B. Cardiac muscle—branched, striated fibres with central nuclei and intercalated disks. C. Smooth muscle—spindle-shaped, non-striated fibres with centrally located nuclei. Adapted from López [18].

Skeletal muscles are composed of microscopic, specialized filaments arranged in a highly organized structure. The fundamental contractile unit of muscle is the sarcomere, defined as the segment between two Z lines [19]. As illustrated in Figure 3.2, each sarcomere contains two primary types of filaments that interact to produce movement: actin (thin filaments) and myosin (thick filaments) [19]. The myosin filaments are located within the anisotropic regions known as A-bands and contain paddle-like projections that attach to actin filaments, forming cross-bridges that generate force during contraction [19].

During muscle contraction, the cross-bridges formed between myosin and actin filaments utilize energy derived from adenosine triphosphate (ATP) to pull the actin filaments toward the center of the sarcomere [20]. This action results in the shortening of the sarcomere and, consequently, the entire muscle. This process, known as the **sliding filament mechanism**, underlies all forms of muscle contraction—whether for voluntary movements such as walking or involuntary actions like the heartbeat [19, 21].

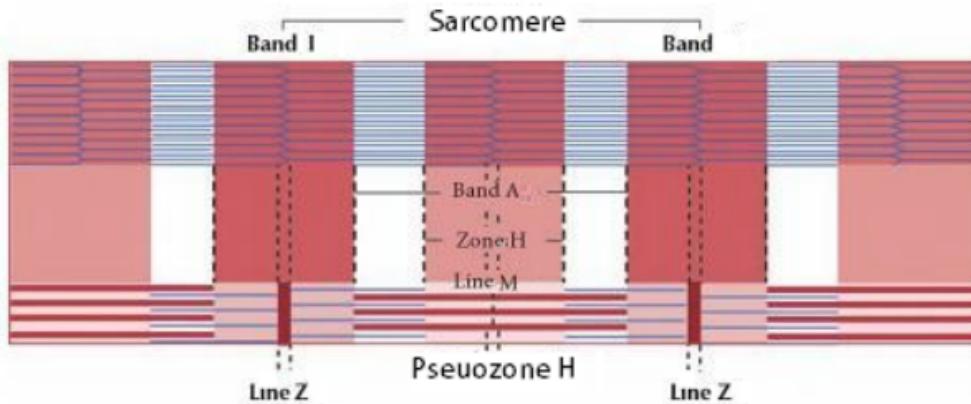


Figure 3.2: Structure of a single sarcomere, the fundamental unit of a muscle fiber. The Z-lines define the boundaries of the sarcomere and anchor the thin actin filaments. The light I-band contains only actin, while the darker A-band shows the region where actin and myosin filaments overlap. At the center of the A-band lies the H-zone, made up of myosin only, and the M-line, which helps stabilize the thick filaments during contraction. Adapted from López [18].

3.2 Action Potentials and Muscle Fiber Activation

Skeletal muscle contraction begins with the generation of action potentials in individual muscle fibers, known as muscle fiber action potentials (MFAPs). These electrical signals propagate along the sarcolemma and trigger contraction within the respective fiber. However, due to their localized and low-amplitude nature, MFAPs contribute only minimally to the surface electromyography (EMG) signal.

In contrast, the motor unit action potential (MUAP) represents the summation of multiple MFAPs from all the muscle fibers innervated by a single α -motor neuron. This neuron, along with the muscle fibers it controls, constitutes a motor unit (MU). When an action potential travels down the motor neuron and reaches the neuromuscular junctions of its associated fibers, it triggers synchronous activation of the entire motor unit. This coordinated firing produces a MUAP, which forms the primary signal component captured in surface EMG recordings. This process is illustrated in Figure 3.3.

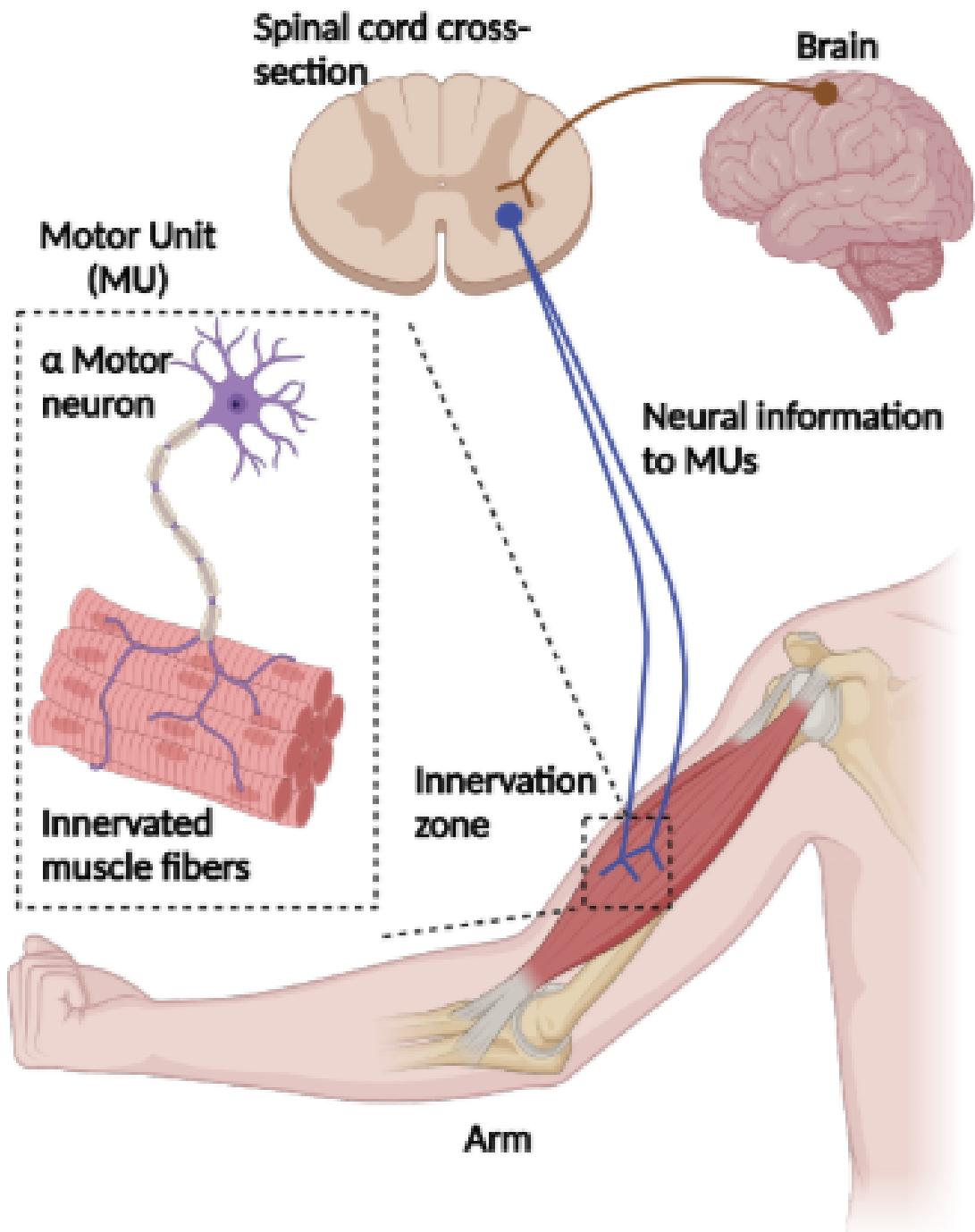


Figure 3.3: Neurophysiology of a motor unit: A neural signal travels from the brain to the spinal cord, reaching an α -motor neuron. The neuron projects to the muscle's innervation zone, where it branches and activates a specific group of muscle fibers. This motor neuron and its fibers form the motor unit. Adapted from Cnejevici [22].

3.3 Resting Membrane Potential

Muscle cells are surrounded by a semipermeable membrane known as the sarcolemma, which maintains a resting membrane potential of approximately **-90 mV** [23]. This potential arises from differences in ion concentrations across the membrane:

- High extracellular concentration of sodium (Na^+)
- High intracellular concentration of potassium (K^+)

These ion gradients are maintained by the selective permeability of the sarcolemma and active transport mechanisms such as the sodium-potassium pump [18, 23]. Resting potential values may vary slightly between fiber types; for instance, slow-twitch fibers tend to exhibit slightly more positive potentials due to increased Na^+ permeability [24].

This process is illustrated in Figure 3.4, which outlines the five key stages of a muscle fiber action potential. The following description is inspired by the works of Kamen and López et al. [18, 23]:

1. **Resting Potential (-90 mV):** At rest, the inside of the muscle cell membrane is negatively charged relative to the outside. Sodium (Na^+) is concentrated outside the cell, while potassium (K^+) is concentrated inside.
2. **Depolarization Initiation:** Upon stimulation, the membrane's permeability to Na^+ increases. If sufficient Na^+ enters and the membrane potential reaches the threshold (approximately -65 mV), voltage-gated sodium channels open, initiating an action potential.
3. **Peak Action Potential (+10 mV):** A rapid influx of Na^+ causes a sharp depolarization, driving the membrane potential up to approximately $+10 \text{ mV}$ —the peak of the action potential.
4. **Refractory Period and Repolarization:** Sodium channels close, halting further Na^+ entry. During the absolute refractory period, the fiber cannot be re-excited. Potassium channels open, allowing K^+ to exit the cell, thereby repolarizing the membrane.
5. **Return to Resting State (-90 mV):** The sodium-potassium pump actively transports Na^+ out of the cell and K^+ back in, restoring the original ion distribution and returning the membrane to its resting potential [18, 23].

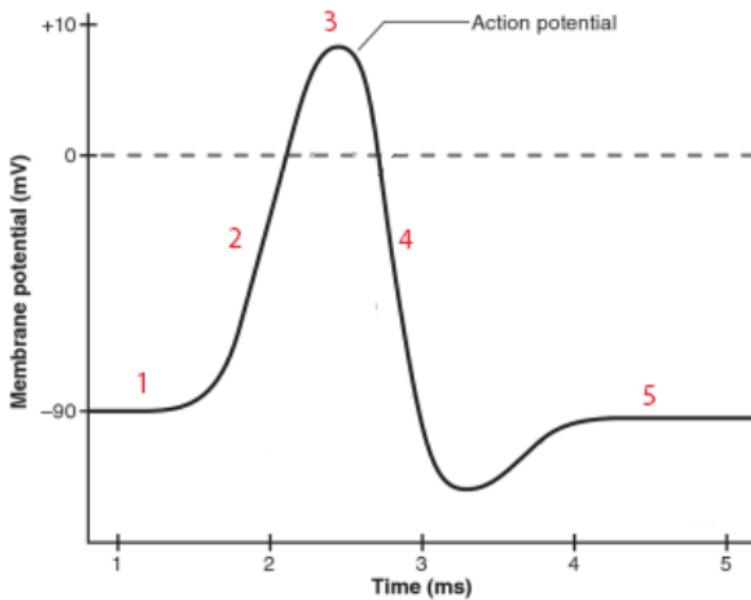


Figure 3.4: Time course of a muscle fibre action potential. At rest (1), the sarcolemma maintains a membrane potential of approximately -90 mV . Upon stimulation (2), increased sodium (Na^+) permeability causes rapid depolarization. The membrane potential peaks around $+10\text{ mV}$ (3), followed by repolarization (4) during the refractory period, as potassium (K^+) exits the cell. Finally, the sodium-potassium pump restores the original ion gradients (5), returning the membrane to its resting potential. Adapted from Kamen [23].

3.4 Motor Unit Physiology and EMG Signal Generation

As illustrated in Figure 3.5, smaller motor units are recruited first, generating slow, low-force muscle twitches. Larger motor units are recruited later to produce stronger and faster contractions. Muscle force output is modulated through two primary mechanisms: the recruitment of additional motor units (motor unit recruitment) and the increase in firing rate of already active units (firing rate modulation).

The surface electromyography (EMG) signal represents the electrical activity associated with muscle contractions and is primarily formed by the summation of action potentials from multiple motor units. Each motor unit generates its own motor unit action potential (MUAP) waveform in response to neural stimulation. These MUAPs occur as pulse trains and are convolved with their characteristic waveform shapes. The summation of all active MUAPs—along with background noise—constitutes the raw surface EMG signal.

As shown in Figure 3.6, this process illustrates how the electrical activity of multiple motor units is combined into a single EMG recording. This composite signal reflects both the spatial and temporal distribution of muscle fiber activation and serves as a foundational metric for assessing neuromuscular function in both research and clinical applications.

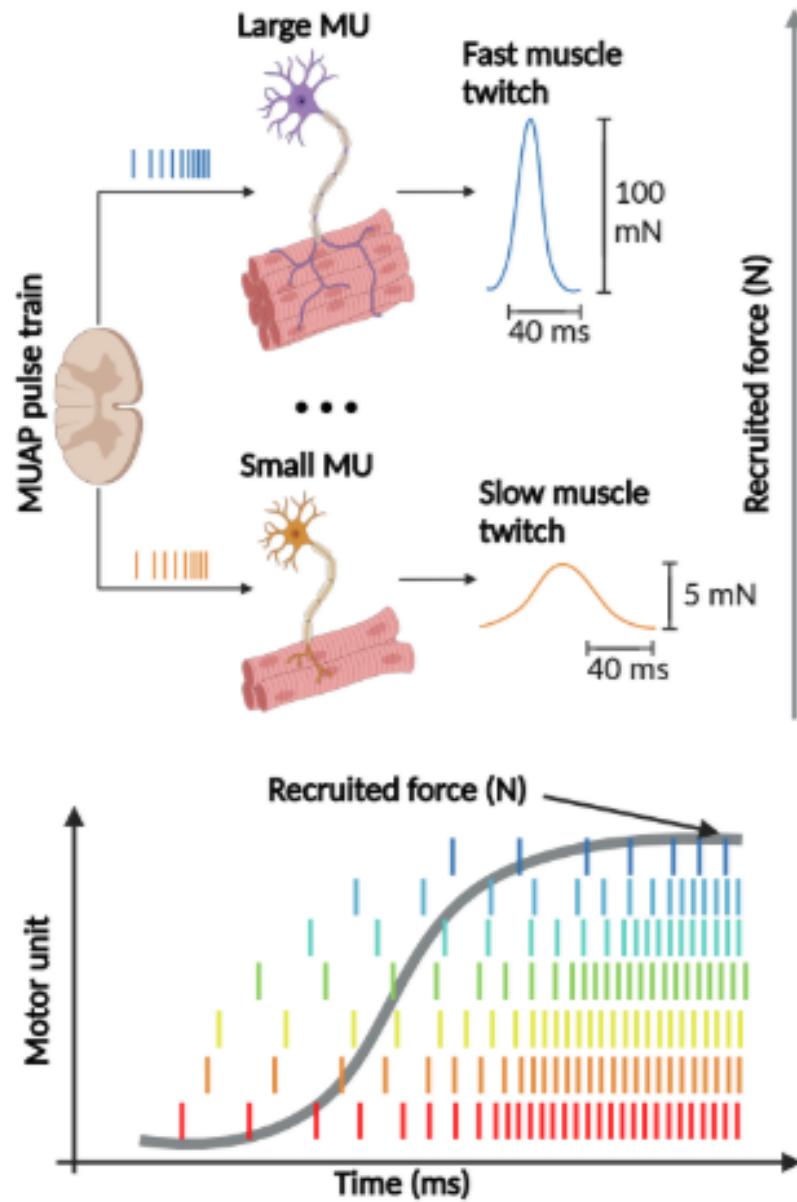


Figure 3.5: Motor unit recruitment and firing rate modulation. Smaller motor units are recruited first, producing slower and lower-force muscle twitches. As muscle demand increases, larger motor units are progressively recruited, generating stronger and faster twitches. Total force output rises through two mechanisms: the sequential recruitment of motor units and the increase in their individual discharge rates, as shown in the lower graph. Adapted from Cnejevici [22].

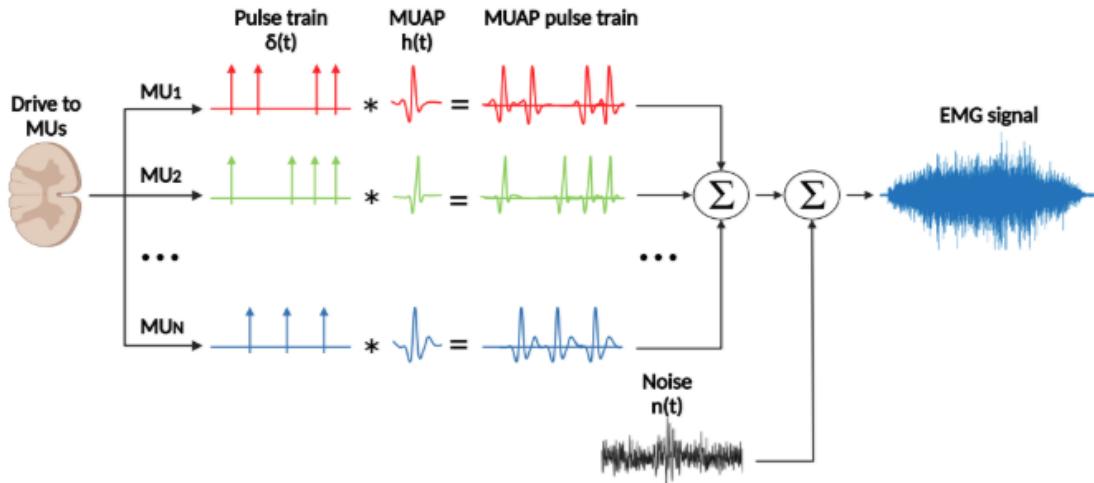


Figure 3.6: Creation of the surface EMG signal. Each motor unit pulse train (drive to motor units) is convolved with its corresponding motor unit action potential (MUAP) waveform. The resulting MUAP pulse trains from multiple motor units are summed together—along with background noise—to form the composite surface EMG signal. This reflects the total electrical activity recorded during muscle activation. Adapted from Cnejevici [22].

3.4.1 Gait and Muscle Synergies

Human bipedal walking is orchestrated by the central nervous system (CNS) through a multilayered hierarchy involving cortical, subcortical, and spinal circuits. Although the outward behavior appears effortless, stable locomotion requires the coordinated activation of dozens of muscles spanning multiple joints. Descending commands from cortical and brainstem centers converge onto spinal pattern-generating networks, where they are integrated with multimodal sensory feedback to produce smooth, energy-efficient gait [25, 26].

Within the spinal cord, rhythmogenic interneuronal networks known as central pattern generators (CPGs) are capable of generating the fundamental oscillatory patterns of gait even in the absence of phasic sensory input. Proprioceptive and cutaneous afferents continuously refine the outputs of these CPGs, while descending brainstem and cortical inputs provide higher-level modulation [25]. Supraspinal areas such as the motor cortex, cerebellum, and basal ganglia further shape locomotor output—enabling real-time balance control and adaptive steering. These structures also assist the CNS in minimizing metabolic cost by leveraging favorable biomechanical dynamics, such as gravity and passive limb pendular motion [26].

A long-standing question in motor neuroscience is how the CNS resolves the so-called degrees-of-freedom problem: how to select from the vast number of possible muscle activation patterns that can achieve the same mechanical outcome [27]. One influential theory posits that the CNS addresses this challenge by recruiting muscle synergies—low-dimensional groupings of muscles that function as single control units [27]. Under this framework, leg muscle activity during gait can be effectively described as weighted combinations of a small number of primitive temporal patterns, despite the underlying complexity of the musculoskeletal system.

3.4.2 Muscle Synergies with Non-Negative Matrix Factorization

Non-negative matrix factorization (NMF) is an algorithmic approach used to extract muscle synergies from EMG data. A key characteristic of NMF is that it allows only additive, not subtractive, combinations of factors. This makes it particularly well-suited for analyzing EMG signals, which are inherently non-negative. Lee et al. [28] introduced NMF as a method for learning parts-based representations of data, and it has since been widely adopted for muscle synergy analysis [29].

In the context of muscle synergies, NMF assumes that the EMG data matrix V , with dimensions m muscles by n time points, can be approximated by the product of two non-negative matrices W and H [30]:

$$V \approx W \times H \quad (3.1)$$

Each column of matrix W represents a fixed pattern indicating how strongly each muscle contributes to a specific synergy. Each row of matrix H describes the time-varying activation of the corresponding synergy.

The number of synergies (r) is typically selected such that the product $W \times H$ explains at least 90% of the variance in the original EMG data [30]. This results in a compact set of muscle synergies along with their corresponding activation timings.

NMF-based analysis has been widely used in locomotor studies to identify synergies that correspond to functional phases of gait. For example, Clark et al. [8] identified synergies associated with weight acceptance, propulsion, swing initiation, and swing. These findings suggest that the central nervous system (CNS) orchestrates gait by recruiting a small number of time-shifted muscle groupings that simplify motor control.

3.4.3 Phases of the Human Gait Cycle and Joint Kinetics

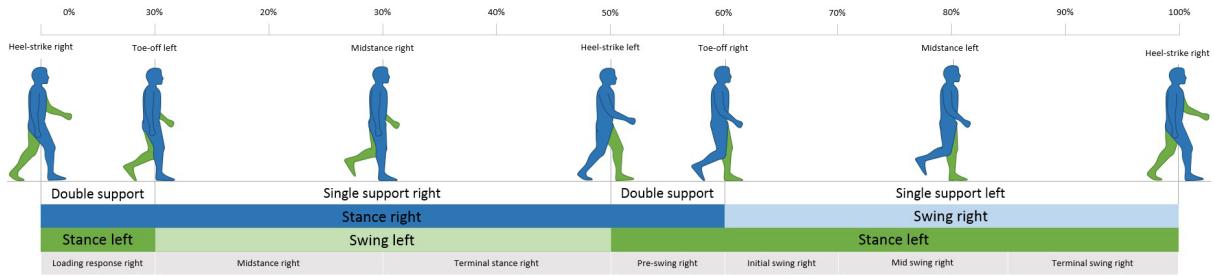


Figure 3.7: Visual representation of the human gait cycle with corresponding limb positions, support phases (double vs. single), and stride sub-phases. Each segment is aligned with the percentage of gait cycle progression and identifies key events such as heel strike and toe-off. Adapted from Kopelovich [31].

Gait Phase (%)	Hip ROM	Knee ROM	Ankle ROM
Initial Contact (0%)	20° flexion	0° extension	0° (neutral)
Loading Response (0–10%)	15° flexion	15° flexion	0–5° plantarflexion
Midstance (10–30%)	0° (neutral)	5° flexion	5° dorsiflexion
Terminal Stance (30–50%)	10–20° hyperextension	0° extension	0° (neutral)
Preswing (50–60%)	10–20° hyperextension	30° flexion	20° plantarflexion
Initial Swing (60–75%)	20° flexion	60° flexion	10° plantarflexion
Midswing (75–85%)	30° flexion	30° flexion	0° (neutral)
Terminal Swing (85–100%)	30° flexion	0° extension	0° (neutral)

Table 1: Range of motion (ROM) requirements at the hip, knee, and ankle joints during each sub-phase of the gait cycle.

The human gait cycle is classically divided into eight sub-phases, spanning from Initial Contact to Terminal Swing. Each phase has a distinct function and kinetic profile governed by the interplay of ground reaction forces, internal torques, and neuromuscular control. While joint configurations vary across the gait cycle, their required range of motion values are listed in Table 1 and illustrated in Figure 3.7.

Initial Contact:

Initial Contact occurs at 0% of the gait cycle and was formerly referred to as heel strike. However, in some pathological gait patterns, the heel may not be the first point of contact with the ground [32]. The primary function of this phase is to establish surface contact and initiate weight acceptance [33].

Kinetically, the lateral aspect of the calcaneus contacts the ground first, and the Ground Reaction Force (GRF) is slightly posterior to the ankle joint, creating a plantarflexion moment. This is opposed by dorsiflexor torque from the tibialis anterior, extensor digitorum longus, and extensor hallucis longus [34]. At the knee, the GRF lies anterior to the joint axis, inducing an extension torque, which is resisted by hamstring activity. At the hip, the GRF is anterior, producing an anterior rotational moment, counteracted by the gluteal musculature [34].

Loading Response:

Loading Response spans 0–10% of the gait cycle and represents the first period of double-limb support. The primary function is shock absorption, continued weight acceptance, and forward progression [33].

During this phase, the center of pressure remains at the posterior calcaneus, and the GRF remains posterior to the ankle, generating a plantarflexion moment. Dorsiflexors act eccentrically to control foot descent [34]. At the knee, the GRF shifts posterior, producing a flexion torque, resisted by eccentric quadriceps activity. At the hip, the GRF remains anterior, creating an external flexion torque and anterior pelvic tilt, controlled by the gluteal extensors [34].

Midstance:

Midstance occurs between 10–30% of the gait cycle, beginning when the contralateral foot lifts off the ground. The primary function is single-limb support and forward progression [33].

Here, the GRF shifts anterior to the ankle, producing an external dorsiflexion moment, resisted by eccentric plantarflexor activity [34]. At the knee, the GRF lies anterior, causing an extension torque. This is controlled by knee flexors (hamstrings) to stabilize the joint as the femur advances over the tibia [34]. At the hip, the GRF is posterior, inducing a posterior pelvic tilt and extension moment, counteracted by hip flexors [34].

Terminal Stance (Heel Off):

Terminal Stance occurs between 30–50% of the gait cycle, beginning with heel rise. The primary function is propulsion and continued single-limb support [33].

At the ankle, the GRF is anterior to the joint, generating an external dorsiflexion moment. This is resisted by eccentric action of the plantarflexors [34]. At the knee, the GRF remains anterior, producing an extension torque, countered by a flexor moment from the hamstrings. At the hip, the GRF lies posterior, creating an external extension moment, opposed by an internal flexor torque [34]. According to Neumann et al. [35], internal torque peaks at this phase include plantarflexion torque at the ankle, flexor torque at the knee, and flexor torque at the hip.

Preswing (Toe Off):

Preswing occurs between 50–60% of the gait cycle and marks the second phase of double-limb support. The function is to deliver the final burst of propulsion and initiate swing [33].

The center of pressure and GRF remain anterior to the ankle joint, resulting in an external dorsiflexion moment, counteracted by internal plantarflexion via gastrocnemius and soleus [34]. At the knee, the GRF lies posterior, creating a flexion torque, resisted by internal extensor torque from the quadriceps. At the hip, the GRF remains posterior, causing a posterior pelvic tilt and extension torque, which is resisted by the hip flexors (iliopsoas) [34].

Initial Swing:

Initial Swing spans 60–75% of the gait cycle, beginning once the foot leaves the ground. Its primary function is to advance the limb and clear the foot [33].

While GRF no longer applies, active muscle contraction continues. At the ankle, dorsiflexors contract concentrically to lift the foot. The knee flexors initiate rapid flexion, while hip flexors advance the limb [32]. These motions ensure sufficient clearance and limb shortening in preparation for Mid Swing.

Mid Swing:

Mid Swing occurs from 75–85% of the gait cycle, ending when the tibia reaches vertical alignment. The function is to maintain foot clearance and continue limb advancement [33].

Ankle dorsiflexors maintain a neutral position, while the knee extends partially through eccentric contraction of the hamstrings, and the hip continues to flex concentrically [32]. Although GRF is absent, internal moments remain critical for controlled progression.

Terminal Swing:

Terminal Swing covers 85–100% of the gait cycle, ending with the next Initial Contact. The main function is to decelerate the limb and prepare for ground contact [33].

Dorsiflexors hold the foot in neutral; the knee extends fully and is stabilized by eccentric then concentric hamstring action. At the hip, flexors complete swing phase, and gluteal extensors engage to stabilize the trunk for upcoming heel strike [32, 34].

3.5 Deep Learning

Deep learning (DL) models are particularly well-suited for solving complex, non-linear problems—such as predicting gait phases from EMG signals—because feedforward neural networks are universal function approximators [36]. In this thesis, we apply deep learning techniques to analyze muscle activation patterns during human locomotion.

Although DL models are capable of learning directly from raw EMG signals, we adopt a feature-based approach to enhance interpretability and training efficiency. Specifically, conventional time-domain features—such as root mean square (RMS), mean absolute value (MAV), waveform length (WL), average amplitude change (AAC), and integrated absolute value (IAV)—are first extracted from the EMG signals. Non-negative matrix factorization (NMF) is then applied to derive muscle synergy activations, which serve as inputs to the deep learning models.

The goal of the DL models in this thesis is to identify temporal patterns in the synergy activations and associate them with specific gait events, such as heel strike and toe-off. These models are designed to learn both spatial relationships across EMG channels and temporal dependencies throughout the gait cycle.

Three deep learning architectures are implemented in this work:

- **One-Dimensional Convolutional Neural Networks (1D CNNs):** Used to extract local activation patterns from the synergy signals.
- **Bidirectional Long Short-Term Memory (BiLSTM) Networks:** Used to capture sequential dependencies in the gait data, processing both past and future temporal contexts.
- **Transformer-Based Models:** Used to model long-range temporal dependencies through self-attention mechanisms, following the architecture introduced by Vaswani et al. [37].

While DL models offer high predictive performance, they require considerable training time and large amounts of labeled data to generalize effectively and avoid overfitting.

4 Methods

4.1 Data Acquisition

To accurately capture both muscle activity and lower-limb movement during walking, two experimental setups were employed: overground walking and treadmill walking. Overground walking refers to natural, self-paced locomotion across a flat surface without the use of a treadmill, more closely resembling everyday walking. In both conditions, high-density surface electromyography (HD-sEMG)—a technique that utilizes closely spaced electrode arrays to record electrical activity from localized muscle regions—was recorded concurrently with marker-based motion capture. All systems were synchronized in real time to ensure precise temporal alignment between neuromuscular and kinematic data.

A total of seven healthy adult participants took part in the study. Four completed the overground walking condition (mean age: 27.75 ± 1.71 years), and three completed the treadmill walking condition (mean age: 26.6 ± 1.7 years).

The study was conducted in accordance with the Declaration of Helsinki and was approved by the Ethics Commission of Friedrich-Alexander University Erlangen-Nürnberg (application numbers: 24-510-S and 24-286-B). All participants provided written informed consent prior to participation, confirming their voluntary involvement and understanding of the study procedures.

Experimental Conditions:

- **Treadmill Walking:** Participants walked at a fixed speed of 0.55 m/s with a cadence of approximately 73 steps per minute (bpm). The treadmill was operated using D-Flow 3 software (Motekforce Link B.V., Amsterdam, The Netherlands), which enabled real-time synchronization of kinematic and sEMG data.
- **Overground Walking:** Participants walked at a natural, self-selected pace, averaging approximately 70 bpm. As with the treadmill condition, sEMG and motion capture data were acquired in parallel.

Each participant completed 23 full gait cycles per condition, corresponding to approximately 120 seconds of continuous walking.

Instrumentation:

- **Kinematic and Kinetic Data:**

For overground walking, marker trajectories were recorded using an OptiTrack motion capture system (NaturalPoint, Inc., Corvallis, Oregon, USA) equipped with PrimeX cameras operating at a sampling rate of 125 Hz. Reflective markers were placed on the right lower limb according to the Rizzoli Lower Body Protocol [38]. Data acquisition was managed using Motive software (NaturalPoint, Inc., Corvallis, Oregon, USA). For treadmill walking, marker data were sampled at 100 Hz, and ground reaction forces were recorded at 1000 Hz using integrated force plates. All data streams were synchronized in real time using D-Flow 3 software (Motekforce Link B.V., Amsterdam, The Netherlands).

- **Surface EMG (sEMG) Data Collection:**

sEMG signals were recorded from the right leg using four Muovi probes (OT Bioelettronica S.r.l., Turin, Italy), each comprising 32 electrodes embedded in high-density EMG (HD-EMG) bracelets (HD20MM1602B). Electrode placement was as follows:

- **Probe 1** – Extensor hallucis longus and soleus: inferior right calf (19–21 cm circumference)
- **Probe 2** – Tibialis anterior: superior right calf (approximately 33 cm circumference)
- **Probe 3** – Vastus lateralis (quadriceps): inferior ventral thigh (approximately 33 cm circumference)
- **Probe 4** – Biceps femoris (hamstrings): superior dorsal thigh (19–21 cm circumference)

The specific probe configurations used for each participant and walking condition are detailed in Table 2.

sEMG signals were high-pass filtered at 10.5 Hz and low-pass filtered at 500 Hz. All channels were sampled at 2000 Hz and digitized at 16-bit resolution using a 24-bit sigma-delta analog-to-digital converter.

Participant ID	Walking Condition	Probe Configuration
001	Overground Walking	Probes 1–4 applied
002	Overground Walking	Probes 1–4 applied
003	Overground Walking	Probes 1–4 applied
004	Overground Walking	Probes 1–4 applied
005	Treadmill Walking	Probes 1–4 applied
006	Treadmill Walking	Probes 2–4 applied; Probe 1 excluded due to technical issues
007	Treadmill Walking	Probes 2–4 applied; Probe 1 excluded due to technical issues

Table 2: Participant allocation across walking conditions and corresponding high-density EMG probe configurations.

Synchronization:

All experimental data streams—including sEMG, motion capture, and force plate recordings—were synchronized using an eSync 2 hub (OptiTrack, NaturalPoint, Inc., Corvallis, Oregon, USA) under both experimental conditions. During treadmill walking, full synchronization was managed via D-Flow 3 software (Motekforce Link B.V., Amsterdam, The Netherlands), ensuring precise temporal alignment across all modalities.

4.1.1 Marker Data Processing

- **Missing Value Handling and Trial Segmentation:** Missing values in the raw marker trajectories were addressed using linear interpolation. A trigger signal was then used to define the onset and offset of each walking trial, and all data outside these temporal boundaries were excluded from further analysis.
- **Spatial Recentering:** Marker coordinates were spatially recentered relative to a designated reference marker. For overground walking trials, which followed the Rizzoli marker set, the RCA marker was used as the reference [38]. For treadmill walking trials recorded with the Vicon Nexus system (Vicon Motion Systems Ltd., Yarnton, Oxfordshire, United Kingdom), the RHEE marker served as the reference. At each time point, the coordinates of the reference marker were subtracted from those of all other markers to normalize spatial positions.

- **Resampling and Normalization:** The cleaned and recentered marker data—originally sampled at 120 Hz (overground) and 100 Hz (treadmill)—were resampled to 2000 Hz using polyphase filtering to match the sEMG sampling rate. Z-score normalization was then applied to ensure consistency and comparability across participants and trials.

4.1.2 Electromyography Data Alignment and Preprocessing

Following synchronization via the eSync 2 system described above, each EMG dataset was processed through a four-stage preprocessing pipeline:

- **Alignment to Trigger Signal:** The EMG signal from each probe was temporally aligned with the corresponding kinematic data using the shared trigger signal. Only the segments within the trigger-defined time window were retained to ensure temporal consistency with the motion capture data.
- **Artifact Removal and Signal Reconstruction:** Z-score normalization was first applied to each EMG channel. Channels with sufficient variance were processed using independent component analysis (ICA) [39] to separate neural signals from noise artifacts. The reconstructed signal components were then re-normalized. Channels with negligible variance were reconstructed using the mean of the retained ICA components to ensure consistency across the dataset.
- **Temporal Interpolation:** The cleaned EMG signals were interpolated onto the same time base as the resampled marker data (2000 Hz), enabling precise temporal alignment between neuromuscular and kinematic features.
- **Data Aggregation:** All preprocessed channels from the four probes were merged into a single unified dataset for downstream analysis.

4.1.3 Feature Extraction from Surface Electromyography Signals

Following synchronization, the combined sEMG data were segmented to extract time-domain features that characterize muscle activation during gait.

Filtering and Windowing:

- **Filtering:**

A 4th-order Butterworth bandpass filter (20–450 Hz) was applied to each channel using forward–backward (zero-phase) filtering to eliminate low-frequency drift and high-frequency noise.

- **Windowing:**

The filtered data were segmented using a 0.1-second window, corresponding to 200 samples at the 2000 Hz sampling rate. A rolling window approach with 99% overlap was used to ensure continuous and temporally dense feature extraction.

Time-Domain Feature Computation

For each window, the following five features were computed:

- **Root Mean Square (RMS):**

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (4.1)$$

- **Waveform Length (WL):**

$$\text{WL} = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (4.2)$$

- **Mean Absolute Value (MAV):**

$$\text{MAV} = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (4.3)$$

- **Integrated Absolute Value (IAV):**

$$\text{IAV} = \sum_{i=1}^N |x_i| \quad (4.4)$$

- **Average Amplitude Change (AAC):**

$$\text{AAC} = \frac{1}{N-1} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (4.5)$$

Here, x_i denotes the sEMG amplitude at sample i , and N is the number of samples in the window. These features have proven effective in capturing muscle activation patterns during gait [40].

The five selected features offer physiological interpretability, low computational complexity, and robustness for EMG-based movement prediction tasks [40]. Specifically, RMS and MAV represent the amplitude of muscle activation, serving as reliable proxies for motor unit recruitment and muscle force. IAV emphasizes the cumulative intensity of the rectified signal, highlighting sustained muscle contractions. WL and AAC capture rapid signal changes and subtle transitions in activation patterns, which are essential for detecting gait phase dynamics and minimizing the effects of electrode displacement. Together, this feature set provides a compact yet comprehensive representation of neuromuscular activity, suitable for downstream synergy decomposition via non-negative matrix factorization (NMF) and classification via deep learning models.

Normalization and Data Refinement:

To standardize feature scales, min–max normalization was applied. Zero values were initially replaced with NaNs and subsequently filled using linear interpolation. An absolute value transformation was then applied to ensure all feature values remained positive. The resulting feature matrix, containing per-window descriptors for each EMG channel, was saved for further analysis.

Motivation for Using Time-Domain Features Instead of Raw Electromyography for Non-Negative Matrix Factorization:

Time-domain features were selected over raw EMG signals for three key physiological and practical reasons. First, NMF requires its input matrix to contain only non-negative values. Because raw EMG signals oscillate around zero, they must be rectified and smoothed—commonly via RMS or MAV—to satisfy this constraint [41]. Second, envelope-based features such as RMS and MAV yield interpretable estimates of muscle activation intensity and have been shown to outperform raw EMG signals in synergy-driven classification tasks [41]. Third, windowed feature extraction substantially reduces noise and data dimensionality, resulting in a more stable input space for matrix factorization.

While Günay et al. [41] used only RMS and MAV in their NMF-based pipeline, our approach incorporates five complementary descriptors—RMS, MAV, WL, AAC, and IAV. These features are physiologically meaningful and computationally efficient, and have been successfully employed in EMG-based movement analysis [40]. In this study, we adopt this expanded feature set to evaluate its suitability for synergy extraction via NMF and for downstream gait-phase modeling.

4.1.4 Muscle Synergy Extraction via Non-negative Matrix Factorization

The methodology, including all formulas and implementation strategies described below, is inspired by prior works [8, 28, 29, 42, 43].

After preprocessing and feature extraction, the full-wave rectified and normalized sEMG data were used to extract muscle synergies. All signals were made strictly non-negative, as required by NMF. Non-negative matrix factorization (NMF) was performed using a custom implementation in PyTorch [44], employing multiplicative update rules to decompose the EMG data matrix $X \in \mathbb{R}^{T \times C}$, where T is the number of time points and C is the number of channels, into two matrices, as shown in Equation 4.6:

$$X \approx WH \quad (4.6)$$

Here, $W \in \mathbb{R}^{T \times K}$ represents the temporal activations of $K = 5$ synergies, and $H \in \mathbb{R}^{K \times C}$ encodes their spatial weights across channels. The multiplicative updates were iteratively computed according to the rules defined in Equation 4.7:

$$H \leftarrow H \circ \frac{W^\top X}{W^\top WH + \epsilon}, \quad W \leftarrow W \circ \frac{XH^\top}{WHH^\top + \epsilon} \quad (4.7)$$

where \circ denotes elementwise multiplication, and ϵ is a small constant to avoid division by zero. The factorization was run for a maximum of 500 iterations.

To evaluate the quality of matrix reconstruction, the percentage of explained variance was computed using Equation 4.8:

$$\text{Explained Variance} = \left(1 - \frac{\|X - \hat{X}\|_F^2}{\|X - \bar{X}\|_F^2} \right) \times 100 \quad (4.8)$$

where $\hat{X} = WH$ is the reconstructed EMG matrix and \bar{X} is the mean of X . To address artifacts such as zero or missing values, the synergy activations were post-processed using cubic spline interpolation. Interpolation was applied only when valid neighboring data points were available, and the resulting values were rectified to preserve the non-negativity constraint.

EMG Channel Selection Strategies:

To reduce the dimensionality of the EMG data and identify the most informative channels for reconstructing synergy activations, three selection strategies were implemented:

1. **Anatomical Selection:** This approach focused specifically on channels located over the vastus lateralis (VL) muscle, recorded using Probe 3 on the inferior ventral thigh. Based on the HD-sEMG layout, the electrodes corresponding to the VL region were predefined. Using RMS values from the feature extraction stage, four channels were consistently selected across all participants for anatomical synergy reconstruction.
2. **Variance-Based Selection:** The variance of each EMG channel within Probe 3 was computed. The top 10 channels with the highest variance—presumed to capture the most dynamic activity from the VL—were selected for further analysis.
3. **L1-Regularized Feature Selection:** A linear model with L1 regularization was trained to predict synergy activations using all channels from Probe 3. The absolute values of the model weights were averaged across all synergy outputs, and the top 10 channels with the highest average weights were selected. This method highlighted the most predictive VL channels based on sparse regression.

Neural Network-Based Mapping and Electromyography Signals Reconstruction:

To assess the quality of each channel selection method, an enhanced neural network was trained to map the reduced EMG signals from the selected channels to the original synergy activations. The objective was to evaluate how effectively the reduced inputs could reconstruct the full synergy dynamics.

The architecture of the network consisted of the following components:

- Three fully connected (dense) layers with 128, 64, and 5 output neurons, respectively,
- Batch normalization and ReLU activation functions after each hidden layer to improve convergence and introduce non-linearity,
- Dropout (rate = 0.2) after each hidden layer to mitigate overfitting,
- A final ReLU activation to enforce non-negativity in the predicted synergy activations.

The network was trained using mean squared error (MSE) loss between the predicted and ground-truth synergy activations. The Adam optimizer was used, and a ReduceLROnPlateau learning rate scheduler was applied to dynamically reduce the learning rate based on validation loss [45]. Each model was trained for up to 2,000 epochs or until the loss plateaued.

After training, the predicted synergy activations \hat{W} were used to reconstruct the full EMG signal, as defined in Equation 4.9:

$$\hat{X}_{\text{NN}} = \hat{W}H \quad (4.9)$$

The reconstruction quality was evaluated using the explained variance metric (Equation 4.8). This enabled direct comparison of reconstruction performance across the anatomical, variance-based, and L1-regularized channel selection strategies.

Upon obtaining the predicted activations \hat{W} , these are multiplied by the spatial weight matrix H to reconstruct the full EMG matrix, after which the explained variance against the original data X is computed to compare channel-selection strategies. The reconstructed EMG corresponding to the optimal method is then retained for all downstream analyses.

Evaluation and Final Selection:

For each channel selection strategy, the output of the neural network was used to reconstruct the full EMG matrix. The quality of the reconstruction was quantified using the explained variance metric, as previously defined in Equation 4.8. The method that achieved the highest explained variance was selected as the optimal channel reduction strategy for subsequent synergy-based gait analysis.

Convolutional Neural Network–Bidirectional Long Short-Term Memory Reconstruction of Tibialis Anterior from Vastus Lateralis:

To evaluate how accurately the activation of the vastus lateralis (VL) can predict that of the tibialis anterior (TA), we trained a hybrid convolutional–recurrent regression model on reconstructed EMG signals. An 80/20 train–test split was applied before temporal windowing to avoid data leakage.

Preprocessing:

- **Channel Selection:** The input matrix \mathbf{X} consisted of reconstructed VL channels, and the target matrix \mathbf{Y} consisted of reconstructed TA channels.

- **Temporal Windowing:** For each time index t , a centered window of length $W = 31$ samples was extracted, resulting in:

$$\mathbf{X} \in \mathbb{R}^{N \times 31 \times C_{\text{in}}}, \quad \mathbf{Y} \in \mathbb{R}^{N \times C_{\text{out}}},$$

where C_{in} and C_{out} are the numbers of input and target channels, respectively.

Network Architecture:

- **Convolutional Encoder:** Two 1D convolutional layers (kernel size = 3, padding = 1) expanded the input from $C_{\text{in}} = 21 \rightarrow 64 \rightarrow 128$. Each layer was followed by ReLU activation and batch normalization.
- **Bidirectional LSTM:** A two-layer BiLSTM with hidden size 64 and dropout rate 0.3 processed the 31-step temporal sequence, producing a 128-dimensional context vector.
- **Linear Readout:** A fully connected layer projected the context vector to the target space of size C_{out} , corresponding to the TA channels.

Training Procedure:

- **Epochs:** The model was trained for 100 epochs.
- **Objective and Optimizer:** Mean squared error (MSE) loss was minimized using the AdamW optimizer.
- **Learning Rate Scheduling:** A One-Cycle learning rate policy was employed for dynamic adjustment.
- **Stability Enhancements:** Training used automatic mixed precision (AMP) and gradient clipping for numerical stability.
- **Batching:** Mini-batches of size 64 were used, with random shuffling at each epoch.

Inference and Evaluation:

- For each output channel i , Pearson correlation r_i was computed between the true and predicted activations:

$$r_i = \text{corr}(Y_{:,i}, \hat{Y}_{:,i}), \quad i = 1, \dots, C_{\text{out}}.$$

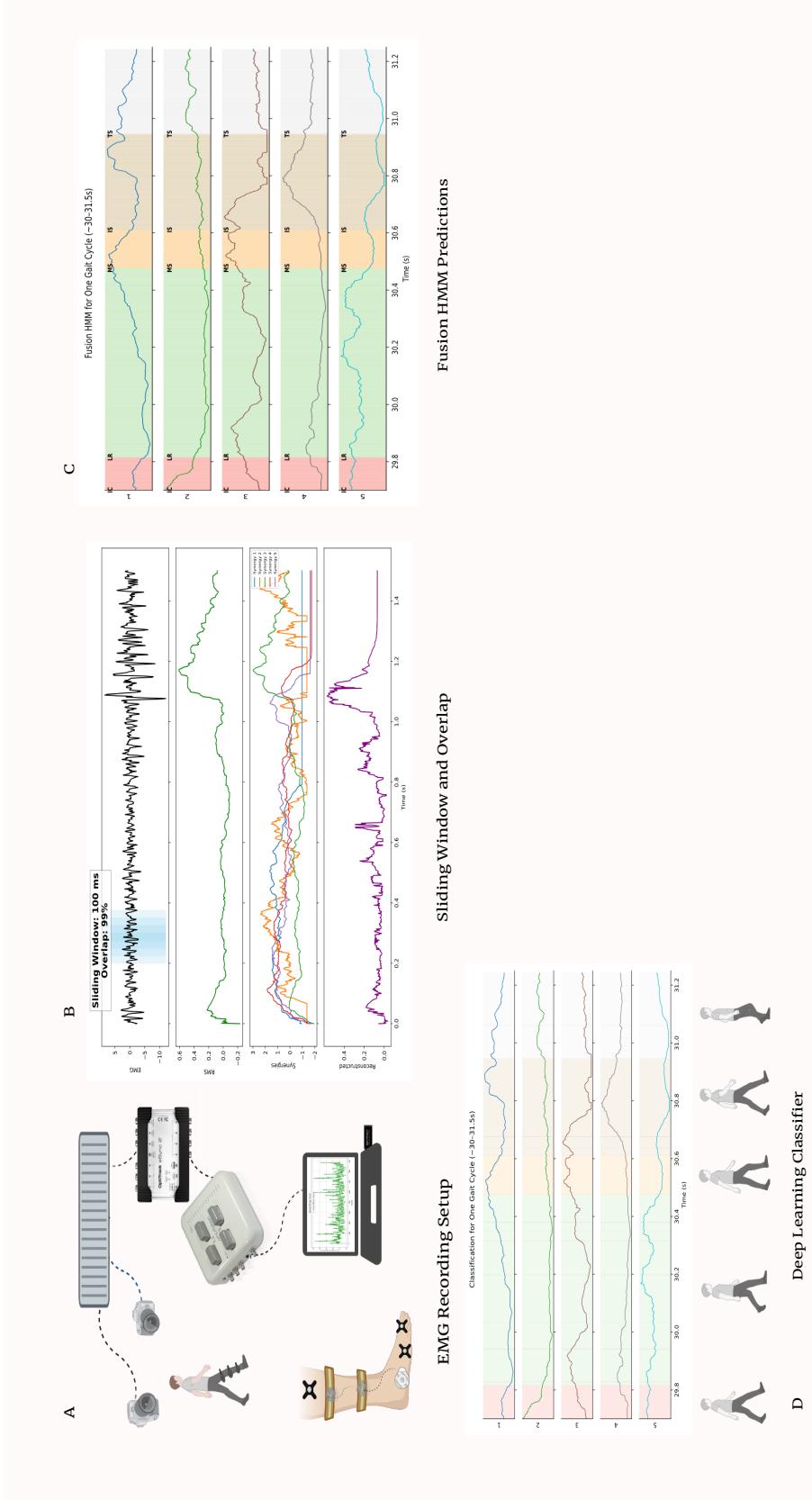


Figure 4.1: (A) High-density surface EMG (HD-sEMG) signals are recorded from lower-limb muscles and data are wirelessly transmitted for real-time monitoring and storage. (B) The EMG signals are processed using a 100 ms sliding window with 99% overlap to extract temporal features and compute muscle synergies via non-negative matrix factorization (NMF); the resulting synergy activation coefficients are then used to reconstruct the original EMG signals, enhancing interpretability. (C) A fusion Hidden Markov Model (HMM), trained on both marker-derived kinematic features and EMG-based synergy activations, is used to infer gait phase labels for each frame within the stride, producing robust temporal segmentation using muscle synergies. (D) A deep learning model-selected as the best performer among BiLSTM, CNN, and Transformer architectures-is trained on the synergy features to classify gait stages with high accuracy, supporting real-time phase decoding for neuroprosthetic control.

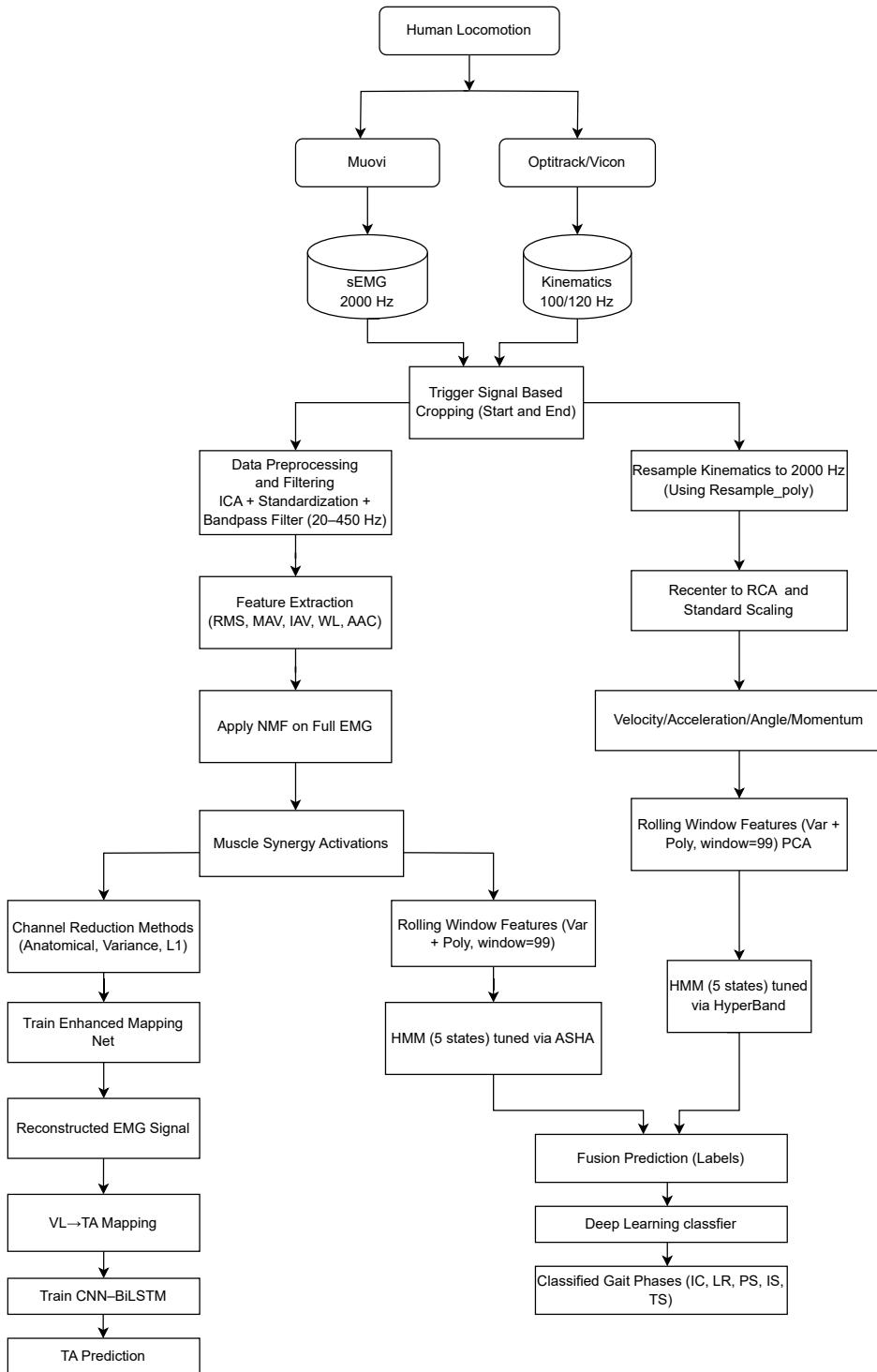


Figure 4.2: Gait phase classification pipeline. The left branch processes EMG signals through feature extraction, synergy analysis, and reduced-channel reconstruction, enabling CNN–BiLSTM-based prediction of tibialis anterior (TA) from vastus lateralis (VL). The right branch handles marker-based kinematics, extracting velocity and momentum features. Both branches yield statistical features used to train HMMs for gait segmentation. Outputs are fused and classified into gait phases: Loading Response (LR), Mid Stance (MS), Terminal Stance (TS), Pre Swing (PS), and Terminal Swing (TS). Adapted from Cnejevici [22].

4.2 Kinematic Feature Extraction

Raw three-dimensional marker trajectories,

$$\mathbf{x}_i(t) \in \mathbb{R}^3,$$

were recorded during gait trials and serve as the basis for kinematic analysis. In our pipeline, these trajectories are processed to derive descriptive features that capture the motion dynamics of anatomical landmarks. The procedure includes:

- Differentiation to compute velocities and accelerations
- Estimation of joint flexion angles via dot products
- Calculation of angular momentum using cross products
- Filtering and statistical characterization over defined time windows

4.2.1 Differentiation: Linear Velocity and Acceleration

To estimate instantaneous velocity and acceleration, position trajectories were numerically differentiated with respect to time. The linear velocity of marker i was computed using the second-order central difference method, as described by Winter [46]:

$$v_i = \frac{x_{i+1} - x_{i-1}}{2\Delta t} \quad (4.10)$$

Similarly, linear acceleration was computed using the central difference approximation:

$$a_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2} \quad (4.11)$$

These calculations were performed independently for each marker and spatial axis (X, Y, Z). The time step Δt represents the uniform temporal interval between successive motion capture frames. This numerical differentiation approach follows standard central difference schemes commonly applied to discrete kinematic data [46].

4.2.2 Joint Flexion Angles via Dot Product

To quantify joint flexion—such as at the knee or hip—angles were computed using the geometric relationship between three anatomical landmarks:

$$\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3 \in \mathbb{R}^3,$$

where \mathbf{p}_2 represents the joint vertex. Two vectors were defined from these points:

$$\mathbf{A} = \mathbf{p}_1 - \mathbf{p}_2, \quad \mathbf{B} = \mathbf{p}_3 - \mathbf{p}_2.$$

The angle θ between vectors \mathbf{A} and \mathbf{B} was computed using the dot product relation, as shown in Equation 4.12:

$$\theta(t) = \cos^{-1} \left(\frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \right) \quad (4.12)$$

where $\|\mathbf{A}\| = \sqrt{a_1^2 + a_2^2 + a_3^2}$, and similarly for $\|\mathbf{B}\|$.

This vector-based formulation for calculating the angle between adjacent limb segments is mathematically equivalent to the law of cosines and is widely used in three-dimensional biomechanical analyses [47]. The resulting angle θ reflects the internal joint angle in degrees, characterizing the flexed or extended posture of the joint. This provides a consistent and anatomically interpretable measure of joint alignment throughout the gait cycle.

4.2.3 Angular Momentum

To capture localized rotational dynamics of anatomical landmarks during gait, we computed the angular momentum of each marker relative to a fixed reference point, following classical mechanics principles [48]. The angular momentum $\mathbf{H}_{O_i}(t)$ of the i th marker is defined as:

$$\mathbf{H}_{O_i}(t) = \mathbf{x}_i(t) \times m_i \mathbf{v}_i(t) \quad (4.13)$$

where:

- $\mathbf{x}_i(t) \in \mathbb{R}^3$ is the position vector of marker i with respect to a fixed origin,
- $\mathbf{v}_i(t) \in \mathbb{R}^3$ is its linear velocity vector, and
- m_i is the point mass associated with the marker, assumed constant.

This formulation follows the classical definition of angular momentum for a particle, typically expressed as $\mathbf{L} = \mathbf{r} \times m\mathbf{v}$, where \mathbf{r} and \mathbf{v} denote position and velocity vectors, respectively [48]. In our pipeline, a unit mass assumption ($m_i = 1$) was adopted, simplifying the expression to:

$$\mathbf{H}_{O_i}(t) = \mathbf{x}_i(t) \times \mathbf{v}_i(t) \quad (4.14)$$

This vector captures the three-dimensional rotational contribution of each marker over time and enriches the kinematic feature space with dynamic cues that are critical for gait phase differentiation.

4.2.4 Filtering and Noise Reduction

To mitigate high-frequency noise and motion artifacts inherent in kinematic recordings, all derived time series—including marker positions, linear velocities and accelerations, joint angles, and angular momentum vectors—were processed using a zero-phase **fourth-order Butterworth low-pass filter** with a **cutoff frequency of 5 Hz**. This filtering strategy is supported by established gait analysis literature, which shows that the primary frequency components of human walking predominantly fall below 5–6 Hz [46, 49, 50]. The selected cutoff frequency thus ensures preservation of relevant biomechanical content while effectively attenuating high-frequency noise and soft tissue oscillations.

The filter was applied in a **forward–backward** (zero-phase) configuration using bidirectional processing, which eliminates phase shifts introduced by single-direction filtering. This approach ensures that the temporal integrity of gait events—such as initial contact, toe-off, and peak joint excursions—is maintained. Preserving phase accuracy is critical for analyzing temporally dependent features such as joint flexion angles and segmental kinematics. The adopted filtering procedure therefore supports both the fidelity and interpretability of the derived kinematic signals.

4.2.5 Windowing and Statistical Feature Computation

After filtering, the continuous kinematic data were segmented into non-overlapping windows of $W = 99$ samples. Within each window, statistical descriptors were computed for each spatial axis (X, Y, and Z). For example, the variance was calculated as:

$$\text{Var}[f] = \frac{1}{W} \sum_{k=1}^W (f_k - \bar{f})^2 \quad (4.15)$$

The variance quantifies the spread of motion within each window, capturing fluctuations in marker position and velocity that reflect the underlying biomechanical variability of movement. High variance values typically correspond to rapid transitions or unstable phases of the gait cycle, whereas low variance indicates steady-state movement—making it a valuable feature for distinguishing between gait phases.

The second-order polynomial trend coefficient was estimated using ordinary least-squares regression, following the standard formulation described by Montgomery et al. [51]. In addition to these descriptors, inter-marker Euclidean distances and acceleration magnitudes were computed as defined in Equations 4.16 and 4.17, respectively:

$$d_{ij}(t) = \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \quad (4.16)$$

$$a_{\text{mag}}(t) = \|\mathbf{a}_i(t)\| \quad (4.17)$$

These features were subsequently aggregated to form a comprehensive descriptor set, which was then reduced in dimensionality using principal component analysis (PCA), yielding a five-dimensional representation for downstream modeling.

4.3 Integration with Electromyography Muscle Synergies and Model Fusion

In parallel with kinematic feature derivation, statistical features were computed from EMG-derived synergy activations using identical windowing strategies. Specifically, the rolling-window variance and second-order polynomial trend coefficients were extracted from the synergy amplitude signals, yielding a five-dimensional feature representation temporally aligned with the PCA-reduced kinematic features.

Although initial attempts using only NMF-derived synergy activations showed promising low-dimensional representations, these signals remained continuous and exhibited variability across strides. To improve temporal alignment and stabilize inter-cycle consistency, we extracted statistical descriptors that capture local fluctuations (via variance) and temporal curvature (via second-order polynomial fitting). This choice is supported by Agostini et al. [52], who addressed stride-to-stride variability by clustering synergy activations, and by Park et al. [3], who showed that NMF-based features outperform traditional EMG descriptors for gait phase classification. In contrast, Guerra et al. [53] bypassed feature engineering and directly modeled preprocessed EMG with BiLSTM, achieving high accuracy through temporal modeling alone. However, we pursued a layered approach to enhance robustness and interpretability for sequential modeling.

Accordingly, two separate five-state Gaussian Hidden Markov Models (HMMs) were trained independently—one on the kinematic features and one on the synergy-derived statistical features—to capture the latent sequential structure of gait.

Each HMM was initialized with a deterministic state distribution $\pi = [1, 0, 0, 0, 0]$.

indicating that the system always begins in the first gait phase. The transition matrix $\mathbf{A} \in \mathbb{R}^{5 \times 5}$ was defined in a left-to-right (Bakis) structure to enforce temporal order across gait stages, as shown in Equation 4.18:

$$A = \begin{bmatrix} \alpha & 1-\alpha & 0 & 0 & 0 \\ 0 & \beta & 1-\beta & 0 & 0 \\ 0 & 0 & \gamma & 1-\gamma & 0 \\ 0 & 0 & 0 & \delta & 1-\delta \\ 1-\theta & 0 & 0 & 0 & \theta \end{bmatrix} \quad (4.18)$$

This structure enables cyclic progression from terminal stance to initial contact. Emission probabilities for each hidden state were modeled using multivariate Gaussian distributions:

$$p(\mathbf{x}_t | z_t = i) = \mathcal{N}(\mathbf{x}_t | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

with full covariance matrices $\boldsymbol{\Sigma}_i$ regularized to be symmetric and positive definite, consistent with the Baum–Welch training procedure [54].

To optimize the transition parameters $\alpha, \beta, \gamma, \delta, \theta$, the stride-wise distribution error was minimized, as formalized in Equation 4.19:

$$\min_{\alpha, \beta, \gamma, \delta, \theta} \text{DistError}(\alpha, \beta, \gamma, \delta, \theta), \quad (4.19)$$

subject to the constraint:

$$0.05 \leq \alpha, \beta, \gamma, \delta, \theta \leq 0.95.$$

Two separate hyperparameter tuning strategies were employed:

- **Markers HMM:** HyperBand [55]
- **Synergy HMM:** Asynchronous Successive Halving Algorithm (ASHA) [56]

Each scheduler executed 1000 trials across a uniform parameter search space.

Posterior probabilities from both models were computed using the forward–backward algorithm and fused post hoc to yield a unified probabilistic estimate of gait phase:

$$P_{\text{fused}}(z_t) = 0.5 \cdot P_{\text{markers}}(z_t) + 0.5 \cdot P_{\text{synergy}}(z_t).$$

The final predicted phase label at each time point was determined using the maximum a posteriori estimate:

$$\hat{z}_t = \arg \max_i (P_{\text{fused}}(z_t = i)),$$

thereby enabling gait phase classification informed jointly by kinematic and neuromuscular features.

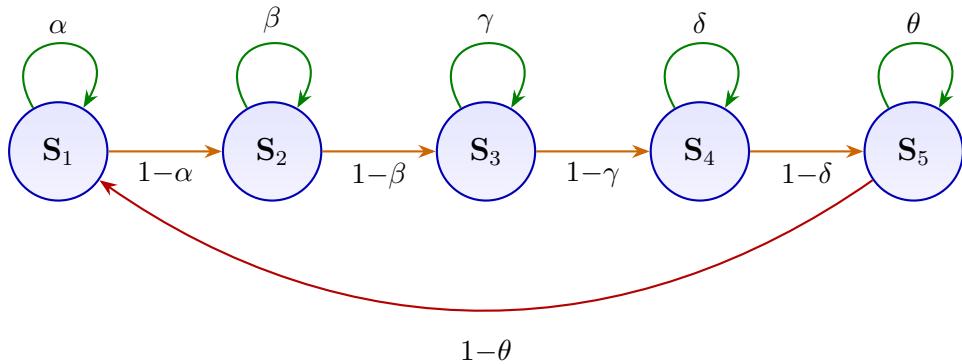


Figure 4.3: Five-state left-to-right (Bakis) Hidden Markov Model with cyclic return. Each state S_i corresponds to a discrete gait phase, with self-loop transitions (green) representing within-phase persistence ($\alpha, \beta, \gamma, \delta, \theta$) and forward transitions (orange) denoting progression through the gait cycle. A cyclic transition (red) from S_5 back to S_1 enables continuous modeling of successive gait strides. Adapted from Roth [13].

4.4 Deep Learning Models for Synergy-Based Gait Phase Prediction

Following the HMM fusion approach that integrated kinematic and neuromuscular signals for gait phase estimation, we employed deep learning models to capture more complex, non-linear temporal dependencies within the data. While HMMs are effective for sequence modeling and provide interpretable state transitions, they are limited by strong assumptions regarding state transitions and observation distributions.

Deep learning models, by contrast, can automatically learn intricate temporal patterns directly from data, without requiring these explicit assumptions. In this study, we utilized EMG-derived muscle synergy signals, segmented into overlapping time windows of fixed length to form multivariate temporal sequences. These sequences reflect the evolving activation patterns over time and were used to train deep learning models in a supervised manner.

Specifically, we trained BiLSTM, CNN, and Transformer architectures using the gait phase labels predicted by the fused HMM model as ground truth. This strategy enabled the deep models to learn directly from the raw synergy time series, eliminating the need for manually engineered features.

The subsections below describe each model architecture, associated training procedures, and the ensemble-based decision fusion strategy employed for final gait phase classification.

4.4.1 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are an advanced type of Recurrent Neural Network (RNN) designed to effectively model sequential data by capturing long-range dependencies. Traditional RNNs often suffer from vanishing or exploding gradient problems, which hinder their ability to retain long-term information. LSTMs overcome these limitations through the introduction of memory cells and a gating mechanism comprising three gates: the input gate, forget gate, and output gate. Each gate plays a critical role in regulating the flow of information through the network.

At each time step, the LSTM processes the current input along with information from the previous time step. Its internal structure determines which information to retain, discard, or output, allowing the network to learn from long temporal sequences and preserve essential information over time.

The theoretical formulation and structure of LSTM presented in this section are inspired by the foundational works of Hochreiter et al. [57], as well as the comprehensive review by Goodfellow et al. [58]. Architectural diagrams and visual illustrations of the gating mechanisms—such as the one shown in Figure 4.4—are adapted from Zhan et al. [59], who proposed a BiLSTM-Attention-based neural network for human gait phase recognition using multi-source data fusion.

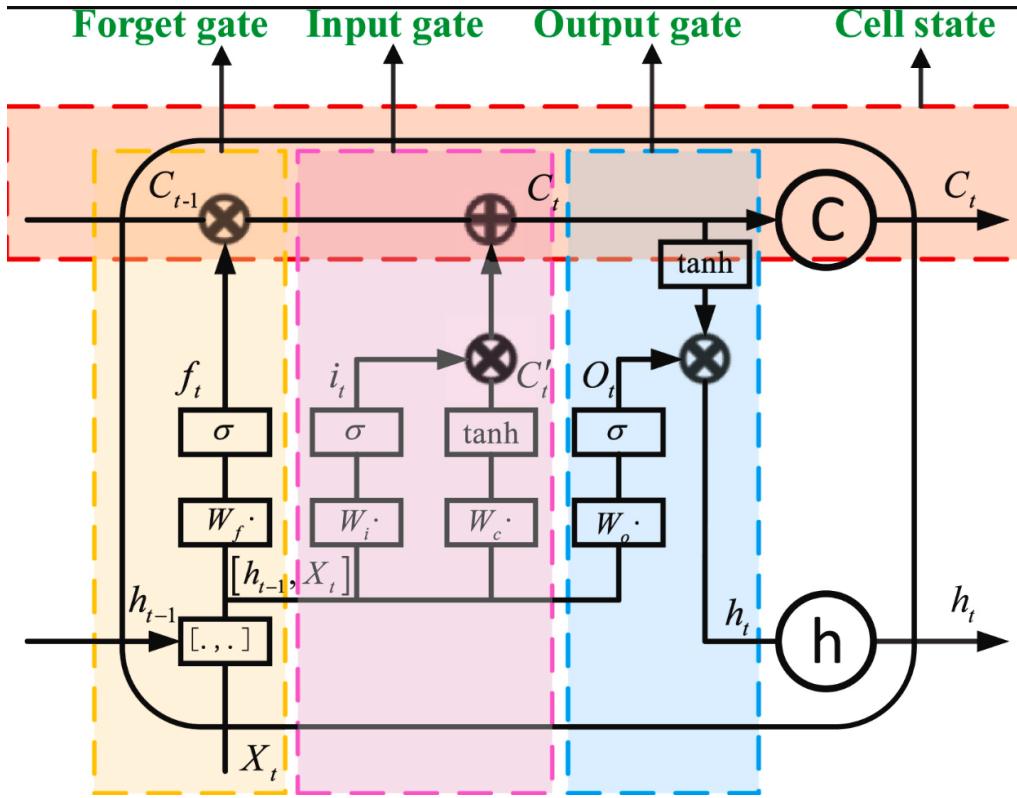


Figure 4.4: LSTM architecture: The Long Short-Term Memory (LSTM) cell consists of three main gates—forget gate, input gate, and output gate—that regulate the flow of information. The forget gate (f_t) controls which parts of the previous cell state (C_{t-1}) are retained. The input gate (i_t) and candidate state (\tilde{C}_t) update the cell state (C_t). The output gate (o_t) determines which parts of the cell state contribute to the output hidden state (h_t). This gated mechanism allows the LSTM to maintain long-term dependencies while mitigating vanishing gradients during training. Adapted from Zhan et al. [59].

- **Input Gate:** Controls what new information should be added to the cell's memory. It decides which parts of the current input and past information are important enough to keep:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (4.20)$$

- **Forget Gate:** Decides what information from the past should be erased from the cell's memory. It helps the model forget things that are no longer important for making future predictions:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (4.21)$$

- **Candidate Cell State:** Suggests new potential information that could be added to the cell's memory, based on the current input and past hidden state. It represents the fresh content the network might want to remember:

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4.22)$$

- **Cell State Update:** Updates the cell's memory by blending the important past information (kept by the forget gate) with the useful new information (chosen by the input gate). This updated memory helps the LSTM retain relevant context over time:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (4.23)$$

- **Output Gate:** Determines how much of the updated cell state should be passed on to the next step. It controls what information from the memory is important enough to influence the output and the next hidden state:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (4.24)$$

- **Hidden State Update:** Generates the final output of the LSTM cell by filtering the updated memory through the output gate. This output carries the most relevant information to the next time step or layer:

$$h_t = o_t \odot \tanh(c_t) \quad (4.25)$$

$\sigma(\cdot)$ – sigmoid activation function, $\tanh(\cdot)$ – hyperbolic tangent activation function,
 \odot – element-wise multiplication

Bidirectional LSTM Networks

Bidirectional Long Short-Term Memory (BiLSTM) networks enhance sequence modeling by processing the input data in both forward and backward directions. Given an input sequence x_1, x_2, \dots, x_T , the model generates a forward hidden state \vec{h}_t by reading the sequence from start to end, and a backward hidden state \overleftarrow{h}_t by reading it from end to start. At each time step t , these two hidden states are concatenated:

$$h_t = [\vec{h}_t; \overleftarrow{h}_t] \quad (4.26)$$

This combined representation captures information from both the past and the future relative to each time step, offering a richer temporal context compared to unidirectional models [60, 61]. By integrating information from both directions, BiLSTMs significantly improve the model's ability to understand sequential dependencies [58].

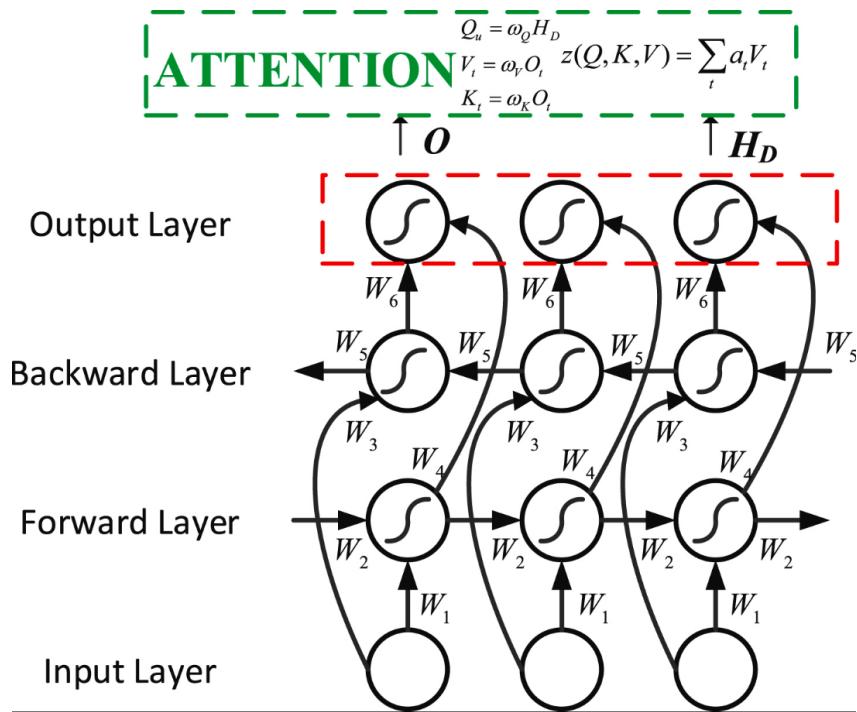


Figure 4.5: Attention layer architecture: The input sequence is first processed by a bidirectional LSTM (BiLSTM), generating forward and backward hidden states that are concatenated and linearly projected into query (Q), key (K), and value (V) vectors—where Q acts as a “question” at each time step, K as an “index” to compare against, and V as the “content” to be aggregated—before being passed to a self-attention layer. Scaled dot-product attention then computes attention weights that highlight the most relevant time steps, and the resulting weighted sum forms a global context vector z that captures the most informative temporal features, which is finally used for classification. Adapted from Zhan et al. [59].

Incorporating Self-Attention

To further improve the model's ability to focus on important moments within a sequence, a custom self-attention mechanism was introduced-drawing inspiration from the scaled dot-product attention proposed by Vaswani et al. [37] and recent adaptations of attention in sequence models by Zhan et al. [59]. Positioned after the BiLSTM layer, this attention mechanism processes the hidden states output as a sequence matrix $H = [h_1, h_2, \dots, h_T]$. The self-attention architecture, visually illustrated in Figure 4.5, dynamically computes importance weights across these hidden states, enabling the model to focus on the most informative portions of the input sequence.

This attention mechanism is implemented in PyTorch, designed to mirror the scaled dot-product attention formulation of Vaswani et al. [37]. It includes trainable projection matrices for queries (W_q), keys (W_k), and values (W_v), and consists of the following steps:

- **Linear Projections:** Hidden states H are projected into three matrices-queries Q , keys K , and values V -via trainable linear transformations:

$$Q = HW_q, \quad K = HW_k, \quad V = HW_v \quad (4.27)$$

- **Attention Weights Computation:** The relevance between queries and keys is assessed by computing scaled dot-product similarity, normalized through the softmax function to obtain the attention weights matrix A :

$$A = \text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) \quad (4.28)$$

- **Context Vector Formation:** A weighted combination of the value vectors is computed at each time step to form intermediate context vectors:

$$Z(t) = \sum_i A(t, i) V_i \quad (4.29)$$

These intermediate vectors are then averaged over all time steps to produce the final context vector:

$$z = \frac{1}{T} \sum_{t=1}^T Z(t) \quad (4.30)$$

The output z is a fixed-length global context vector that encapsulates the most relevant temporal dynamics across the entire sequence. This representation is then passed to the subsequent dense layers for classification.

Output Layer and Probabilities:

The final linear layer yields raw logits, which are fed directly to the cross-entropy loss during training. A separate softmax activation is applied to these logits only when class probabilities are required.

Layer	Input Shape	Output Shape	Function
Input	(B, T, D)	(B, T, D)	Raw input sequence
Bidirectional LSTM 1	(B, T, D)	$(B, T, H \times D_b)$	First LSTM (hidden_size = units_1)
BatchNorm1d	$(B, T, H \times D_b)$	$(B, T, H \times D_b)$	Normalize over feature channels
Dropout	$(B, T, H \times D_b)$	$(B, T, H \times D_b)$	Dropout (rate = dropout_1)
Bidirectional LSTM 2 (optional)	$(B, T, H \times D_b)$	$(B, T, H \times D_b)$	Second LSTM if second_lstm=True
BatchNorm1d (optional)	$(B, T, H \times D_b)$	$(B, T, H \times D_b)$	Normalize after second LSTM
Dropout (optional)	$(B, T, H \times D_b)$	$(B, T, H \times D_b)$	Dropout (rate = dropout_2)
Attention	$(B, T, H \times D_b)$	$(B, H \times D_b)$	Self-attention + mean pooling over time
Linear (FC)	$(B, H \times D_b)$	(B, F)	Fully connected (units = fc_units)
ReLU	(B, F)	(B, F)	Non-linear activation
BatchNorm1d	(B, F)	(B, F)	Normalize after FC
Dropout	(B, F)	(B, F)	Dropout (rate = fc_dropout)
Linear (Output)	(B, F)	(B, C)	Final linear layer \rightarrow raw logits

Table 3: Layer-by-layer breakdown of the proposed BiLSTM-based classification model. The architecture includes optional stacked BiLSTM layers, self-attention with mean pooling, and a feedforward classification head. Tensor dimensions are defined using symbols listed in Table 4.

Symbol	Name	Description
B	Batch size	Number of sequences processed in one batch
T	Sequence length	Number of time steps (elements) per sequence
D	Input feature dimension	Number of features per time step in the input
H	LSTM hidden size	Number of hidden units in the LSTM layer
D_b	Bidirectional factor	2 if LSTM is bidirectional, 1 otherwise
F	Fully connected size	Number of units in the dense (FC) layer
C	Number of classes	Number of output classes for classification

Table 4: Definition of tensor shape symbols used throughout the BiLSTM model description in Table 3.

4.4.2 Convolutional Neural Network (CNN)

While BiLSTMs are designed to capture long-range temporal dependencies, one-dimensional Convolutional Neural Networks (CNNs) are better suited for modeling short-term, local patterns in sequential data-making them well-suited for detecting brief bursts of muscle synergy activity through learnable temporal filters. Convolutional Neural Networks (CNNs), originally developed for image processing tasks [62], have also proven useful for analyzing time-series data. In one-dimensional form, CNNs apply filters along the time axis, allowing them to detect local patterns and short-term dependencies in sequential inputs [63]. This makes them a better choice for modeling structured signals such as EMG [64].

$$X = [x_1, x_2, \dots, x_T], \quad x_t \in \mathbb{R}^d \quad (4.31)$$

where T denotes the length of the sequence and d is the number of features at each time step.

Temporal Convolutions:

Each 1D convolutional layer applies a filter $w \in \mathbb{R}^k$ of kernel size k across local regions of the input [65]. The output feature map y is computed as:

$$y_i = \sigma \left(\sum_{j=0}^{k-1} w_j \cdot x_{i+j} + b \right) \quad (4.32)$$

where b is a learnable bias term, and $\sigma(\cdot)$ is a non-linear activation function such as ReLU or tanh. Padding is applied to preserve the temporal resolution, ensuring that the output has the same length as the input. Multiple such layers can be stacked, allowing the network to capture increasingly abstract temporal features [63].

Global Max Pooling:

After convolutional layers, a global max pooling operation reduces the temporal dimension by selecting the maximum activation across time for each feature map [66]:

$$z = \max_{1 \leq i \leq T} \{y_i\} \quad (4.33)$$

This operation retains the most prominent features and produces a fixed-length output regardless of input sequence length [63].

Fully Connected Layers:

The pooled vector $z \in \mathbb{R}^m$, where m is the number of filters in the final convolutional layer, is passed through one or more dense layers. A dense layer performs:

$$h = \sigma(Wz + b) \quad (4.34)$$

where $W \in \mathbb{R}^{n \times m}$ is a trainable weight matrix, $b \in \mathbb{R}^n$ is a bias vector, and σ is typically ReLU [67]. Dropout may be applied for regularization [63].

Output Layer and Probability Extraction:

The final fully connected layer projects the learned feature vector $h \in \mathbb{R}^F$ into raw class scores (logits) $z \in \mathbb{R}^C$ via

$$z = W_{\text{out}} h + b_{\text{out}}, \quad z = [z_1, \dots, z_C]. \quad (4.35)$$

To obtain a valid probability distribution over the C classes without compromising numerical stability during training, these logits are left unnormalized in the main forward pass. When required for inference or evaluation, the softmax transform is applied externally:

$$\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)}, \quad \hat{y} = [\hat{y}_1, \dots, \hat{y}_C]. \quad (4.36)$$

This separation ensures that the model trains on raw logits—benefiting from greater numerical stability—while still allowing convenient extraction of interpretable class probabilities as needed.

Layer	Input Shape	Output Shape	Function
Input	(B, T, D)	(B, T, D)	Read raw input sequence.
Permute to Channels	(B, T, D)	(B, D, T)	Reorder to (batch, channels, time).
Repeat for each conv block $i = 0 \dots N - 1$:			
Conv1D_i	(B, C_{i-1}, T)	(B, C_i, T)	1D conv filters=filters_i,kernel=kernel_size_i
Activation_i	(B, C_i, T)	(B, C_i, T)	Apply activation activation_i.
Dropout_i (optional)	(B, C_i, T)	(B, C_i, T)	Apply dropout if dropout_i=True (rate=dropout_rate_i).
End repeat			
Global Max Pooling	(B, C_N, T)	$(B, C_N, 1)$	Adaptive max-pool to length 1.
Squeeze Time Dim.	$(B, C_N, 1)$	(B, C_N)	Remove singleton time dimension.
Dense	(B, C_N)	(B, F)	Fully connected (units=dense_units).
ReLU	(B, F)	(B, F)	Apply ReLU activation.
Dropout	(B, F)	(B, F)	Apply dropout (rate=dense_dropout).
Classifier (Linear)	(B, F)	(B, C)	Final linear \rightarrow raw logits.

Table 5: Layer-by-layer architecture of the CNN-based classification model. The network processes raw sequential input using stacked 1D convolutional layers with optional dropout and non-linear activation, followed by global max pooling and fully connected layers. The final output layer produces unnormalized class logits. For symbol definitions and hyperparameter explanations, see Table 6.

Symbol	Name	Function
B	Batch size	Number of sequences per batch
T	Sequence length	Number of time steps per sequence
D	Input dimension	Features per time step
C_i	Conv block i channels	Output channels of block i
filters_i	Conv1D filters	Number of filters in block i
kernel_size_i	Kernel size	Width of Conv1D kernel in block i
activation_i	Activation	“relu” or “tanh” in block i
dropout_i	Dropout flag	Enable dropout in block i
dropout_rate_i	Dropout rate	Drop probability in block i
F	Dense units	Units in final FC layer
dense_units	FC layer size	Number of units in FC layer
dense_dropout	FC dropout rate	Drop probability after FC layer
C	Classes	Number of output categories

Table 6: Definitions of symbols, shapes, and hyperparameters used throughout the CNN-based model description in Table 5.

4.4.3 Transformer Network

Transformer encoders analyze full sequences in parallel, making them well-suited for modeling global temporal relationships, unlike BiLSTMs and CNNs. Transformer [37], initially introduced for natural language processing tasks, have recently been adapted for a variety of time-series learning problems due to their capability to model long-range dependencies using self-attention mechanisms. Unlike traditional architectures such as Long Short-Term Memory (BiLSTM) networks [57] or Convolutional Neural Networks (CNNs) [68], which primarily capture sequential or local patterns, Transformers operate in parallel and compute attention scores to dynamically weigh interactions between all positions in the input sequence.

In this implementation, the input tensor of shape $(T \times d)$, where T is the number of time steps and d is the number of features, is first permuted to $(d \times T)$ to match the input requirements of PyTorch’s 1D convolutional layer [44]. A standard 1D convolution is then applied along the temporal dimension using filters that operate across all input feature channels. This process captures local temporal patterns while also allowing interaction between features, projecting the multi-channel input into a shared embedding space of dimension d_{embed} . The resulting representation of shape $(T \times d_{\text{embed}})$ serves as input to the Transformer encoder.

Furthermore, the schematic diagram shown in Figure 4.6 is adapted based on the architectural layout presented by Carneros-Prado et al. [69], tailored here for our biomedical time-series classification task. The final classification layer produces raw logits, which are subsequently converted into class probabilities using the softmax function during inference or evaluation.

Figure 4.6, adapted from Carneros-Prado et al. [69], keeps the original post-norm layout for visual clarity, but our implementation differs in four respects. First, every encoder block follows the pre-norm order (LayerNorm \rightarrow sub-layer \rightarrow residual), which improves gradient stability in deep stacks. Second, the sinusoidal positional encoding implied in the figure is replaced by learned positional embeddings realised as a trainable lookup table. Third, once the encoder stack is complete we apply an explicit global-pooling stage—mean, max, or first-token—selected with the pool_type hyper-parameter. Finally, the single Fully Connect box is instantiated as a two-layer MLP head (Linear \rightarrow ReLU \rightarrow Dropout \rightarrow Linear), and softmax is applied only during inference. These adjustments align the implementation with current best practice while preserving the conceptual flow illustrated in the diagram.

Let the input sequence be denoted as:

$$X = [x_1, x_2, \dots, x_T], \quad x_t \in \mathbb{R}^d \quad (4.37)$$

where T is the number of time steps and d is the number of features at each step.

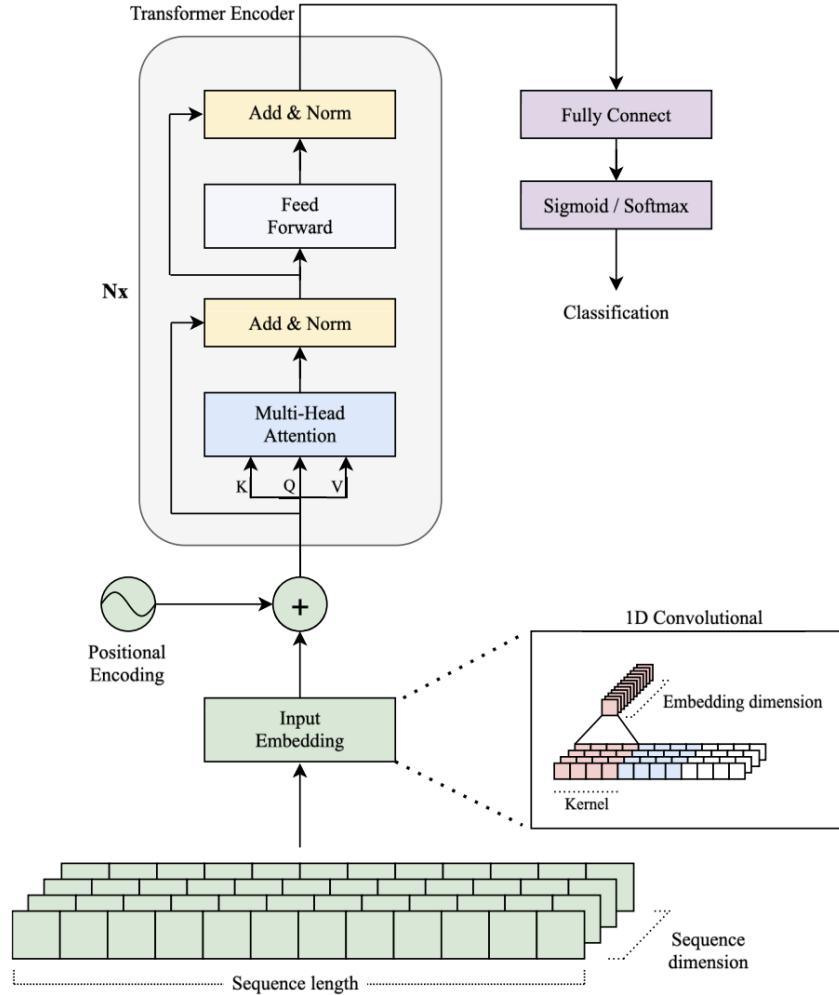


Figure 4.6: Overview of the Transformer-based time-series classification model. The raw input sequence is first embedded via a 1D convolutional layer and enriched with learned positional encodings. The resulting sequence is processed by N pre-norm Transformer encoder blocks—each consisting of multi-head self-attention, a two-layer feed-forward network with ReLU and dropout, residual connections, and layer normalization. The final hidden states are aggregated (mean, max, or first-step pooling) into a fixed-length context vector, which is passed through a ReLU-activated dense layer with dropout and a linear classifier to produce logits for softmax or sigmoid probability estimation. Adapted from Carneros-Prado et al. [69]

Stacked Transformer Blocks ($N \times$):

In the schematic Figure 4.6, adapted from Carneros-Prado et al. [69], each Transformer encoder block is depicted in the classic post-norm arrangement—attention followed by residual addition and layer normalization, then a feedforward network followed by another residual and normalization. In our implementation, however, we employ the pre-norm variant: each sub-layer (multi-head self-attention or feedforward network) is preceded by layer normalization, and the residual connection is applied afterward. This ordering has been shown to improve gradient stability and convergence when stacking deeper encoder layers.

Multi-Head Self-Attention:

A key component of the Transformer is the self-attention mechanism. Given an input sequence, the self-attention layer projects it into three matrices: queries Q , keys K , and values V via learned linear transformations:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \quad (4.38)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d_k}$ are trainable weight matrices, and d_k is the attention dimension [37].

The scaled dot-product attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4.39)$$

Multi-head attention repeats this mechanism h times with different projections and concatenates the results:

$$\text{MHA}(X) = [\text{head}_1; \dots; \text{head}_h]W^O \quad (4.40)$$

where $W^O \in \mathbb{R}^{hd_k \times d}$ is a learned output projection.

Feedforward Network (FFN):

After attention, a position-wise feedforward network is applied to each time step:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (4.41)$$

where W_1, W_2 and b_1, b_2 are learnable weights and biases. Dropout is applied for regularization. Layer normalization is used in the residual connection to ensure stable gradient propagation [70, 71].

Residual Connections and Normalization:

Residual connections are applied after both attention and FFN blocks:

$$\text{Output} = \text{LayerNorm}(x + \text{SubLayer}(x)) \quad (4.42)$$

This helps in training deep networks by preserving gradient flow.

Global Feature Aggregation

After the final Transformer encoder block, we obtain a sequence of hidden states

$$\{\mathbf{h}_t \in \mathbb{R}^{d_{\text{embed}}}\}_{t=1}^T.$$

These are aggregated into a single context vector $\mathbf{z} \in \mathbb{R}^{d_{\text{embed}}}$ via

$$\mathbf{z} = \begin{cases} \frac{1}{T} \sum_{t=1}^T \mathbf{h}_t, & \text{if pool_type = mean} \\ \max_{1 \leq t \leq T} \mathbf{h}_t, & \text{if pool_type = max} \\ \mathbf{h}_1, & \text{if pool_type = first} \end{cases} \quad (4.43)$$

This allows summarizing the entire time series by average, maximum, or first-step pooling.

Classification Head and Probability Computation:

The pooled vector \mathbf{z} is first passed through a ReLU-activated dense layer with dropout and then linearly projected to raw logits $\ell \in \mathbb{R}^C$:

$$\mathbf{u} = ((W_{\text{head}} \mathbf{z} + b_{\text{head}})), \quad (4.44)$$

$$\ell = W_{\text{class}} \mathbf{u} + b_{\text{class}}. \quad (4.45)$$

A ReLU activation is applied after the dense layer to introduce non-linearity before applying dropout and final linear projection.

These unnormalized scores are the direct output of the model. Class probabilities are obtained by

$$p_i = \frac{\exp(\ell_i)}{\sum_{j=1}^C \exp(\ell_j)}, \quad i = 1, \dots, C, \quad (4.46)$$

ensuring a valid probability distribution over the C classes.

Layer	Input Shape	Output Shape	Function
Permute for Conv1D	(B, T, D)	(B, D, T)	Reorder to (batch, channels, time)
Conv1D embedding	(B, D, T)	$(B, embed_dim, T)$	1D conv (out_channels=embed_dim kerel=kernel_size)
ReLU	$(B, embed_dim, T)$	$(B, embed_dim, T)$	Apply ReLU activation
Permute back	$(B, embed_dim, T)$	$(B, T, embed_dim)$	Reorder to (batch, time, embed_dim)
LearnedPositionalEncoding	$(B, T, embed_dim)$	$(B, T, embed_dim)$	Add learned positional embeddings
Repeat for each TransformerBlock $i = 1 \dots N$:			
TransformerBlock_i	$(B, T, embed_dim)$	$(B, T, embed_dim)$	Self-attention + FFN + residual + LayerNorm (heads=num_heads, ff_dim=ff_dim, dropout=block_dropout)
End repeat			
Global pooling	$(B, T, embed_dim)$	$(B, embed_dim)$	Pool by pool_type (mean, max, or first)
Dense	$(B, embed_dim)$	$(B, embed_dim)$	Fullyconnected (units=embed_dim)
ReLU	$(B, embed_dim)$	$(B, embed_dim)$	Apply ReLU activation
Dropout	$(B, embed_dim)$	$(B, embed_dim)$	Apply dropout (rate=head_dropout)
Classifier (Linear)	$(B, embed_dim)$	(B, C)	Linear \rightarrow raw logits.

Table 7: Layer-by-layer architecture of the Transformer-based time-series classification model. The network embeds the input sequence using a 1D convolutional layer, augments it with learned positional encodings, and processes it through N stacked Transformer blocks. Each block includes multi-head self-attention, a feedforward network, residual connections, and layer normalization. Global pooling is used to generate a fixed-length representation, followed by a dense classification head. Symbol definitions are listed in Table 8.

Symbol	Name	Description
B	Batch size	Number of sequences processed in one batch.
T	Sequence length	Number of time steps per sequence.
D	Input feature dimension	Features per time step in the input.
embed_dim	Embedding dimension	Output channels of the Conv1D embedding layer.
num_heads	Attention heads	Heads in each multi-head self-attention block.
ff_dim	Feed-forward size	Hidden dimension of each Transformer FFN layer.
block_dropout	Block dropout rate	Dropout probability inside each TransformerBlock.
pool_type	Global pooling type	One of “mean”, “max” or “first” pooling modes.
head_dropout	Head dropout rate	Dropout probability after the final dense head.
C	Number of classes	Number of output categories for classification.

Table 8: Definitions of tensor dimensions and hyperparameters used in the Transformer-based model (Table 7).

4.5 Output and Ensemble Fusion

Each of the three model architectures—bidirectional long short-term memory (BiLSTM) with attention, convolutional neural network (CNN), and Transformer—produces a raw five-dimensional logit vector at its final fully connected layer. During training, these logits are optimized directly using the cross-entropy loss, which applies the log-softmax internally.

At inference time, each model’s logits are converted into a valid probability distribution by applying the softmax function over the five output dimensions:

$$\mathbf{p}^{(m)} = \text{softmax}(\text{logits}^{(m)}), \quad m \in \{\text{BiLSTM}, \text{CNN}, \text{Transformer}\},$$

so that each $\mathbf{p}^{(m)}$ contains nonnegative entries summing to one.

To improve robustness, the outputs of all models are fused by averaging their softmax probability vectors. If M models produce outputs for a given input window ($M = 2$ or 3),

the ensemble probability is computed as:

$$\mathbf{p}^{(\text{comb})} = \frac{1}{M} \sum_{m=1}^M \mathbf{p}^{(m)}. \quad (4.47)$$

The final gait phase label is obtained by applying an arg max operation to $\mathbf{p}^{(\text{comb})}$. This late-fusion approach—based on softmax outputs rather than raw logits or hard labels—exploits the complementary strengths of the models: temporal memory in BiLSTM, local pattern detection in CNN, and global context awareness in the Transformer. As a result, predictions near phase transitions become more stable.

To determine the best-performing model or ensemble for deployment, each candidate is evaluated on an unseen validation set using three metrics: accuracy, weighted F1-score, and weighted ROC AUC. A combined performance score is then calculated:

$$S = 0.5 \times \text{F1}_{\text{weighted}} + 0.5 \times \text{ROC_AUC}_{\text{weighted}}, \quad (4.48)$$

and the configuration with the highest S is selected.

Its predicted labels are assigned to the center frame of each input window. Any resulting gaps in the frame-level annotations are resolved by forward and backward propagation to produce a continuous gait phase timeline.

4.6 Hyperparameter Tuning

Model performance is highly sensitive to hyperparameter settings. To ensure optimal generalization and robustness, each model was systematically tuned using multiple search strategies within a stratified cross-validation framework. This section outlines the tuning objectives, search algorithms, cross-validation integration, and final training configurations applied to all models.

4.6.1 Hyperparameter Tuning Objective and Cross-Validation Integration

Let $\theta \in \Theta$ denote a configuration of hyperparameters for a given model. The goal is to identify the optimal configuration θ^* that maximizes validation performance, formalized in Equation 4.49:

$$\theta^* = \arg \max_{\theta \in \Theta} f(\theta) \quad (4.49)$$

where the performance function $f(\theta)$ is defined as the validation F1 score:

$$f(\theta) = \text{val_F1_score} \quad (4.50)$$

This objective was evaluated within a three-fold stratified cross-validation setup to ensure consistency and reduce variance in performance estimation across folds.

4.6.2 Search Strategies for Hyperparameter Tuning

Efficient exploration of the hyperparameter space is critical for enhancing model generalization while minimizing computational cost. Three search strategies were used in this study: Bayesian optimization, random search, and Hyperband.

Bayesian Optimization:

Bayesian optimization treats hyperparameter tuning as a black-box optimization problem, where evaluating the performance of each configuration is computationally expensive. It constructs a probabilistic surrogate model that captures both the expected performance and uncertainty associated with different configurations. An acquisition function then selects the next configuration by balancing exploration and exploitation—prioritizing regions likely to yield improvement. This strategy typically achieves superior results with fewer evaluations compared to exhaustive or grid search approaches [72].

Random Search:

Random search samples hyperparameter configurations uniformly at random and retains the best-performing one. While simple, it has proven effective—particularly in high-dimensional spaces where only a subset of parameters significantly affects performance. It is trivially parallelizable and can be interrupted at any time, making it suitable for scenarios with constrained computational resources [73].

Hyperband:

Hyperband extends the principle of successive halving to efficiently allocate computational resources. It begins by evaluating a large number of randomly sampled configurations with minimal resources (e.g., few epochs or small datasets). Poor performers are iteratively discarded, and additional resources are allocated to better-performing candidates. By adaptively balancing exploration (evaluating many configurations) and exploitation (focusing on the most promising ones), Hyperband achieves competitive performance with significantly reduced training time [55].

4.6.3 Hyperparameter Search Spaces for Different Algorithms

To effectively explore the architectural and training choices available for deep learning models, we defined distinct yet bounded hyperparameter spaces for each model family. These spaces were designed to balance expressiveness with computational feasibility and were informed by widely adopted practices in the literature [37, 58, 74].

The selected parameters span architectural design (e.g., number of layers, units, attention heads), training-related variables (e.g., learning rate, dropout), and regularization factors (e.g., ℓ_1 and ℓ_2 penalties). The following tables summarize the search spaces used for the BiLSTM, CNN, and Transformer models, which form the basis of the tuning experiments described earlier.

LSTM-based Model

Hyperparameter	Description	Range / Choices
<code>units_1</code>	Number of units in first LSTM layer	32, 64, 96, 128
<code>bidirectional</code>	Use bidirectional LSTM	True, False
<code>dropout_1</code>	Dropout rate after first LSTM layer	0.1, 0.2, 0.3, 0.4, 0.5
<code>second_lstm</code>	Flag to add a second LSTM layer	True, False
<code>units_2</code>	Units in second LSTM layer (if used)	16, 32, 48, 64
<code>dropout_2</code>	Dropout rate after second LSTM layer	0.1, 0.2, 0.3, 0.4, 0.5
<code>fc_units</code>	Number of units in the Dense layer	32, 64, 96, 128
<code>fc_dropout</code>	Dropout rate after Dense layer	0.2, 0.3, 0.4, 0.5
<code>learning_rate</code>	Learning rate for Adam optimizer	$1e-4 - 1e-2$ (log sampling)

Table 9: Hyperparameter search space for the LSTM-based model. Parameters include the number of LSTM units, bidirectionality, optional second LSTM layers, dropout rates, fully connected layer size, and learning rate.

CNN-based Model

Hyperparameter	Description	Range / Choices
<code>conv_layers</code>	Number of convolutional layers	1, 2, 3
<code>filters_i</code>	Number of filters in layer i	32, 64, 96, 128
<code>kernel_size_i</code>	Convolutional kernel size in layer i	3, 5, 7
<code>activation_i</code>	Activation function in layer i	relu, tanh
<code>dropout_i</code>	Apply dropout after layer i (flag)	True, False
<code>dropout_rate_i</code>	Dropout rate for layer i (if applied)	0.1, 0.2, 0.3, 0.4, 0.5
<code>dense_units</code>	Number of units in Dense layer	64, 128, 192, 256
<code>dense_dropout</code>	Dropout rate after Dense layer	0.1, 0.2, 0.3, 0.4, 0.5

Table 10: Hyperparameter search space for the CNN-based model. This includes the number of convolutional layers, filters, kernel sizes, activation functions, dropout configuration, and the size of the final dense layer.

Transformer-based Model

Hyperparameter	Description	Range / Choices
<code>embed_dim</code>	Embedding dimension for Conv1D layer	64, 128, 256
<code>kernel_size</code>	Kernel size of Conv1D embedding	3, 5, 7
<code>num_blocks</code>	Number of Transformer encoder blocks	2, 3, 4, 5, 6
<code>num_heads</code>	Number of attention heads per block	2, 4, 8
<code>ff_dim</code>	Feed-forward network hidden size in each block	128, 256, 384, 512
<code>block_dropout</code>	Dropout rate inside each Transformer block	0.0, 0.1, 0.2, 0.3
<code>pool_type</code>	Method to pool sequence for classification	Mean, Max, First
<code>head_dropout</code>	Dropout rate on the final dense head	0.0, 0.1, 0.2, 0.3, 0.4, 0.5
<code>learning_rate</code>	Learning rate for the optimizer	1e-5 – 1e-3 (log sampling)

Table 11: Hyperparameter search space for the Transformer-based model. Search dimensions include embedding size, the number of encoder blocks and attention heads, feed-forward network sizes, dropout rates, and final dense layer configuration.

4.6.4 Integration with Cross-Validation

All deep learning models (BiLSTM, CNN, Transformer) were trained on pre-segmented input sequences of fixed length—200 samples (equivalent to 0.1 seconds at 2000 Hz)—with 99% overlap between consecutive windows. This high degree of overlap was used to maximize temporal resolution and improve the continuity of gait-phase transitions.

To obtain robust performance estimates, we incorporated K -fold cross-validation into the hyperparameter tuning process. Specifically, Stratified K -Fold Cross-Validation [75] was employed to preserve class distributions across folds. For each fold k , the training data was split into a training and validation set, and the validation F1 score $f^{(k)}(\theta)$ was computed. The average validation performance for a given hyperparameter configuration was then calculated as:

$$\bar{f}(\theta) = \frac{1}{K} \sum_{k=1}^K f^{(k)}(\theta) \quad (4.51)$$

The optimal configuration θ^* was selected by maximizing $\bar{f}(\theta)$.

To support model comparison across architectures and ensembles, a combined performance score was also defined as the average of the F1 score and the ROC AUC, as shown in Equation 4.52:

$$\text{Combined Score} = 0.5 \times \text{Average F1} + 0.5 \times \text{Average ROC AUC} \quad (4.52)$$

4.6.5 Train–Test Split and Final Evaluation

To ensure fair and unbiased evaluation, the full dataset was partitioned into an 80/20 train–test split using stratified sampling to maintain class balance. The 20% test set was selected from the stride-aligned sequences and held out at the beginning of the pipeline. It remained completely untouched during model development, hyperparameter tuning, and training—thus providing a reliable estimate of generalization performance.

The remaining 80% of the data was used for model development. Within this subset, stratified 3-fold cross-validation was applied for hyperparameter optimization. After tuning, the final models were retrained on the entire oversampled training set using the best hyperparameter configuration obtained for each architecture.

For ensemble evaluation, predictions from the final BiLSTM, CNN, and Transformer models were collected on the held-out test set. Their output probability vectors were averaged, and the class with the highest mean probability was selected as the ensemble prediction. This ensemble was then evaluated using multiple metrics—accuracy, weighted F1 score, and ROC AUC—to determine the best-performing configuration.

4.7 Final Model Selection and Retraining

After identifying the optimal hyperparameters θ^* for each model type (e.g., BiLSTM, CNN, Transformer) through cross-validation, the corresponding model was retrained on the entire oversampled training dataset. This final retraining step ensures that the model benefits from both the full training data and the best-found configuration, thereby maximizing its generalization capacity.

The selection and retraining process followed these guiding principles:

- The validation F1 score served as the primary metric for evaluating model performance.
- Hyperparameter optimization was performed using three strategies: Bayesian Optimization, Random Search, and Hyperband.
- A stratified K -fold cross-validation scheme ensured balanced class distributions across folds and robust performance estimation.
- Final model selection was based on a combined score, defined as the mean of the validation F1 score and the ROC AUC (see Equation 4.52).

The following section describes the implementation of the training procedure using the optimal hyperparameters obtained for each model.

4.8 Training Configuration and Optimization

Following model architecture definition and hyperparameter tuning, each deep learning model was trained using standardized optimization settings. The training procedure is detailed below.

All models were implemented and trained using PyTorch on a workstation equipped with an NVIDIA RTX 4090 GPU. The training process employed the Adam optimizer in conjunction with the CrossEntropyLoss function for multi-class classification. Learning rate scheduling was managed using a ReduceLROnPlateau strategy, configured with a reduction factor of 0.5 and a patience of 1 epoch. Early stopping was applied with a patience of 3 epochs to prevent overfitting, with model weights restored to their best-performing state upon termination.

Each model was trained for a maximum of 10 epochs per fold, with early stopping often halting training sooner when no further improvement in validation loss was observed. This strategy improved convergence efficiency and reduced the risk of overfitting during model development.

A fixed batch size of 16 was used across all training runs. A stratified 3-fold cross-validation strategy was adopted to ensure class balance and robust performance estimation. Hyperparameter tuning was performed using three optimization techniques: Bayesian Optimization, Random Search, and Hyperband.

To address class imbalance, oversampling was applied using either SMOTE or RandomOverSampler, depending on the class distribution in the training set. Input data were segmented into overlapping sequences using a fixed window size and overlap ratio. To improve temporal alignment across samples, Dynamic Time Warping (DTW) was applied to the input sequences.

Three deep learning architectures were trained independently: a BiLSTM model with attention, a one-dimensional CNN, and a Transformer-based model. Ensemble predictions were computed by averaging the predicted class probabilities from each model.

Model interpretability was facilitated using the LIME framework, and GPU device configuration was managed through a custom utility function. To accelerate training and reduce memory consumption on GPU hardware, all models were trained using a mixed-precision strategy. This approach enabled faster computation while maintaining numerical stability during gradient updates.

4.8.1 Loss Function

The loss function for classification was based on categorical cross-entropy and evaluated across epochs using the weighted F1-score and validation loss, as defined in Equation 4.50. All models, including the BiLSTM with attention, 1D CNN, and Transformer, utilize the categorical cross-entropy loss function. This was implemented using CrossEntropyLoss from the PyTorch library. The loss computes the negative log-likelihood between the predicted class probabilities and the true class labels, providing a robust and widely used objective for optimizing classification performance in multi-class settings.

5 Results

5.1 Synergy–Phase Alignment with Gait Subphases

To determine whether each of the five extracted muscle-synergy patterns aligns with a specific gait subphase, the reconstructed muscle activation patterns were first segmented according to validated gait-cycle intervals for each participant. This filtering balanced the number of samples per phase, yielding exactly 92 strides per synergy for overground (normal) walking ($23 \text{ strides} \times 4 \text{ participants}$) and 69 strides per synergy for treadmill walking ($23 \text{ strides} \times 3 \text{ participants}$).

Statistical Overview

Objective: Determine whether each extracted muscle synergy is phase-locked to a single gait subphase and whether that mapping is invariant across walking modes.

Tests:

- **Mann–Whitney U test (two-sided):** Performed separately for each synergy and gait subphase to compare activation distributions between overground and treadmill walking.
- **Benjamini–Hochberg False Discovery Rate (FDR) correction:** Applied to control for multiple comparisons across the 25 synergy–phase combinations.
- **Significance threshold:** Adjusted p-value < 0.05 (FDR level $q = 0.05$).

5.1.1 Statistical Comparison of Synergy Activation Across Walking Conditions

Mann–Whitney U Test with Benjamini–Hochberg Correction

This analysis evaluates whether muscle synergy activations differ between overground and treadmill walking across distinct gait phases. For each synergy and gait cycle phase, a two-sided Mann–Whitney U test was performed to compare activation distributions between walking conditions. To correct for multiple comparisons, the Benjamini–Hochberg False Discovery Rate (FDR) method was applied.

Null hypothesis (H_0): For a given synergy and gait phase, activation levels are identical across walking conditions.

Alternative hypothesis (H_1): At least one synergy–phase combination exhibits significantly different activation levels between conditions.

Test details:

- Total comparisons: 25 (5 synergies \times 5 gait phases)
- Statistical test: Mann–Whitney U (two-sided)
- Correction method: Benjamini–Hochberg FDR ($q=0.05$)
- Significance threshold: Adjusted p-value < 0.05

Unit of analysis. For each synergy–phase combination the 23 strides recorded from a given participant were collapsed to a single summary statistic (median activation). This yielded one observation per participant Overground=4, Treadmill=3. Two-sided Mann–Whitney U tests method=exact were then applied to these participant-level medians, followed by Benjamini–Hochberg FDR correction across the 25 comparisons.

Results Summary:

- FDR-significant combinations: **0 of 25** (0%)
- Non-significant combinations: **25 of 25** (100%)

Synergy	Phase	Significant	Effect Size (r)	Adj. p -value
1	0–10%	No	0.38	0.114
1	10–50%	No	0.19	0.400
1	50–60%	No	0.38	0.114
1	60–80%	No	0.19	0.400
1	80–100%	No	0.38	0.114
2	0–10%	No	0.19	0.400
2	10–50%	No	0.00	0.600
2	50–60%	No	0.38	0.114
2	60–80%	No	0.19	0.400
2	80–100%	No	0.00	0.600
3	0–10%	No	0.19	0.400
3	10–50%	No	0.38	0.114
3	50–60%	No	0.19	0.400
3	60–80%	No	0.00	0.600
3	80–100%	No	0.19	0.400
4	0–10%	No	0.00	0.600
4	10–50%	No	0.19	0.400
4	50–60%	No	0.19	0.400
4	60–80%	No	0.38	0.114
4	80–100%	No	0.00	0.600
5	0–10%	No	0.38	0.114
5	10–50%	No	0.19	0.400
5	50–60%	No	0.00	0.600
5	60–80%	No	0.19	0.400
5	80–100%	No	0.38	0.114

Table 12: Participant-level Mann–Whitney U results for each synergy–phase pair Overground=4, Treadmill=3. No comparison survives Benjamini–Hochberg FDR correction ($q = 0.05$).

These participant-level results indicate that, within the statistical power afforded by seven subjects, muscle-synergy activations do not differ significantly between overground and treadmill walking. The phase-locking patterns themselves (contour plots in Figures 5.1–5.2) remain visually consistent across modes, suggesting a conserved temporal structure even though amplitude differences could not be detected.

Synergy–Phase Mapping

Condition	Initial Contact 0–10%	Mid-Stance 10–50%	Terminal Stance 50–60%	Pre-Swing 60–80%	Terminal Swing 80–100%
Overground	Synergy 2	Synergy 5	Synergy 3	Synergy 4	Synergy 1
Treadmill	Synergy 1	Synergy 4	Synergy 2	Synergy 3	Synergy 5

Table 13: Dominant muscle synergy–gait phase mapping for overground and treadmill walking.

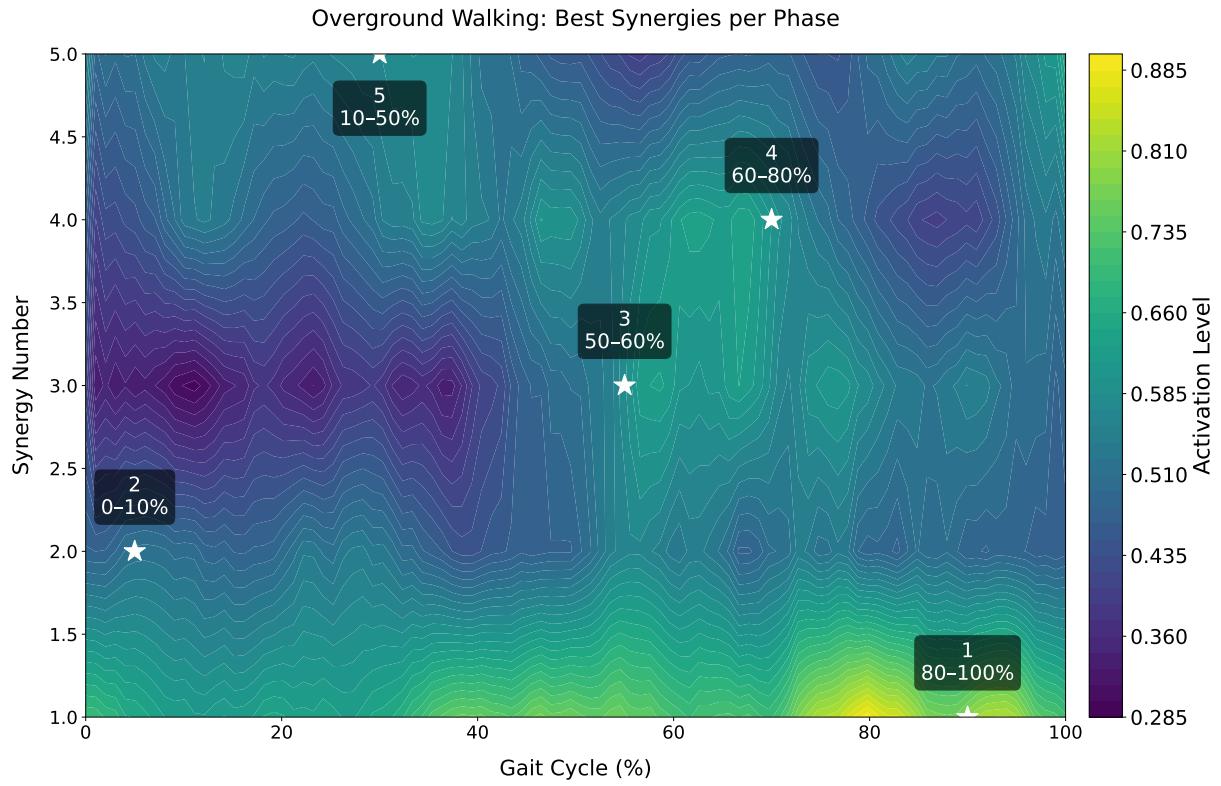


Figure 5.1: Contour plot of synergy activation over the gait cycle for Overground walking. Synergy 1 peaks at 80%–100% (Terminal Swing), Synergy 2 peaks at 0%–10% (Loading Response), Synergy 3 peaks at 50%–60% (Terminal Stance), Synergy 4 peaks at 60%–80% (Pre-Swing), and Synergy 5 peaks at 10%–50% (Mid-Stance).

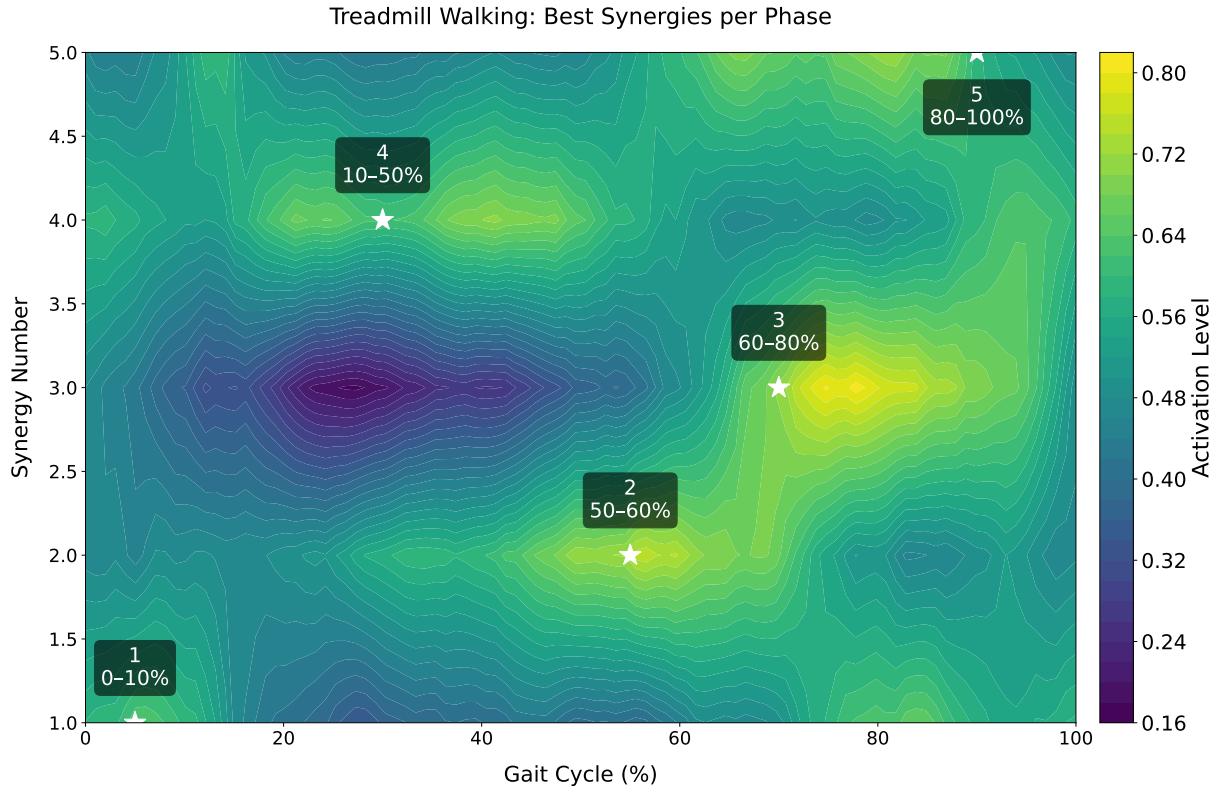


Figure 5.2: Contour plot of synergy activation over the gait cycle for Treadmill walking. Synergy 1 peaks at 0%–10% (Loading Response), Synergy 2 peaks at 50%–60% (Terminal Stance), Synergy 3 peaks at 60%–80% (Pre-Swing), Synergy 4 peaks at 10%–50% (Mid-Stance), and Synergy 5 peaks at 80%–100% (Terminal Swing).

The preceding analyses confirmed that each muscle synergy exhibited a tightly phase-locked activation pattern; however, the identity of the dominant synergy in each gait subphase shifts between overground and treadmill walking (Table 13). Crucially, after Benjamini–Hochberg correction, none of the 25 synergy–phase comparisons showed significant differences in activation magnitude, indicating that the temporal and amplitude structure of these synergies is conserved between conditions. We therefore turned to the issue of dimensionality: quantifying how much additional variance in the electromyographic recordings is captured as the model complexity increases from three to five synergies.

5.2 Statistical Analysis of Muscle Synergy Patterns

Statistical Overview

Objective: Identify the minimum number of synergies required to reconstruct EMG activity with negligible loss of variance.

Tests:

- Friedman test – non-parametric repeated-measures comparison of explained variance for 3, 4, and 5 synergies.
- Elbow/plateau visual inspection – examine variance-explained curves to decide where additional synergies provide diminishing returns.

5.2.1 Friedman Test Analysis

Friedman Test for Synergy Count Differences

This analysis asks whether the explained-variance scores differ across three, four, and five extracted synergies. Because normality cannot be assumed, a Friedman test (the non-parametric analogue of repeated-measures ANOVA) was used. Explained-variance values were first averaged over walking modes for each participant, so each of the seven participants contributed one score at each synergy count.

Null hypothesis (H_0):

The median explained variance is the same for 3, 4, and 5 synergies.

Alternative hypothesis (H_1):

At least one synergy count yields a different median explained variance.

Test results:

- Sample size: $n = 7$ participants (repeated measures)
- Friedman statistic: $\chi^2(2) = 14.0$
- Exact p -value: 0.00091
- Kendall's $W = \frac{14.0}{7 \times 2} = 1.00$ (very large effect)

Since $p < 0.05$ (and W indicates a strong effect), we reject H_0 : explained variance does differ across the three synergy counts. Figures 5.5 and 5.4 show an elbow around four synergies, with only modest gains thereafter.

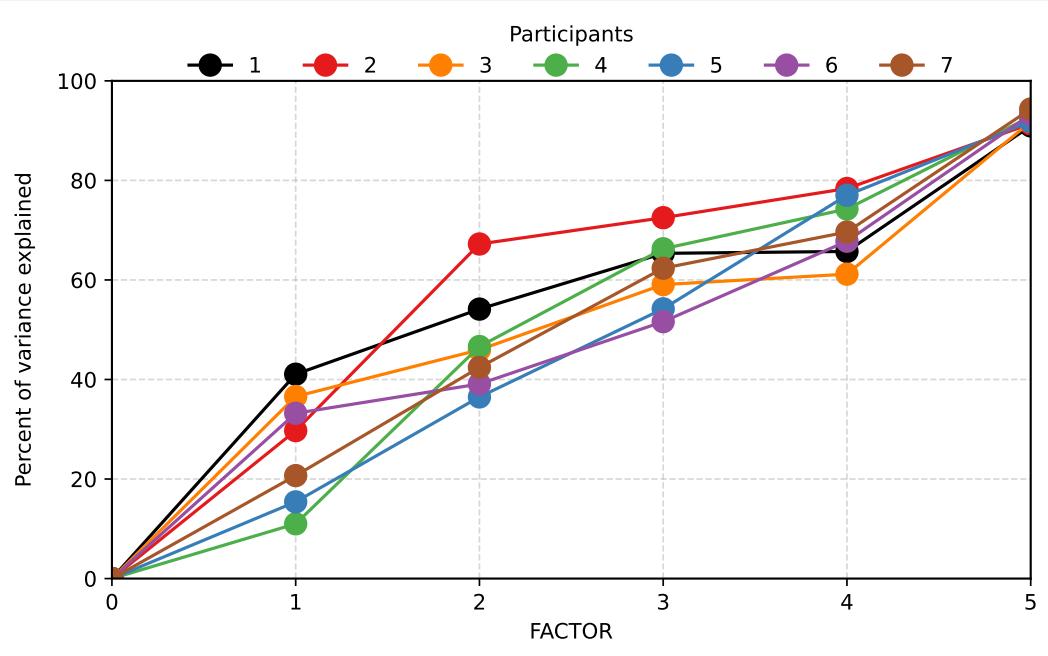


Figure 5.3: Participant-wise variance explained as a function of synergy count. Each line represents one participant’s variance reconstruction performance for 1 to 5 extracted synergies. The increasing trend supports the selection of higher synergy counts, aligning with the Friedman test results.

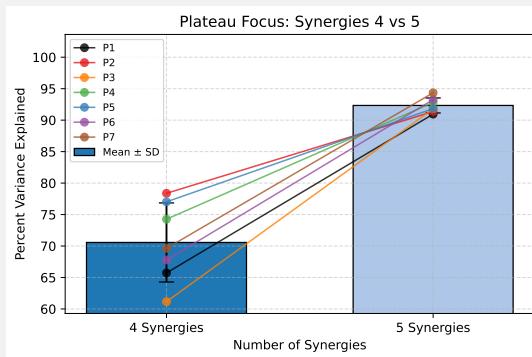


Figure 5.4: Plateau focus on explained variance for 4 vs. 5 synergies. Bars denote the group mean \pm SD, while individual colored lines trace each participant’s increase.

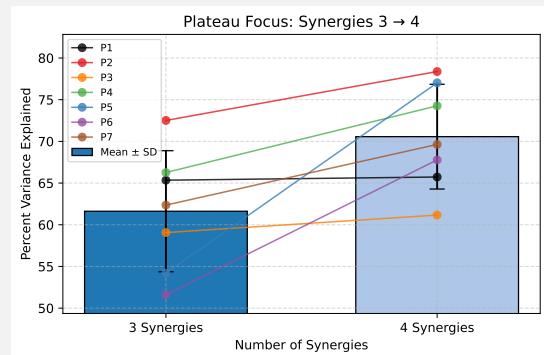


Figure 5.5: Plateau focus on explained variance for 3 vs. 4 synergies. Bars denote the group mean \pm SD, while individual colored lines trace each participant’s increase.

With five synergies confirmed as the minimal set that preserves variance, the subsequent analysis addresses whether reconstruction accuracy depends on the algorithm used to derive those synergies. Accordingly, we compare three methods—Anatomical, ℓ_1 -regularised, and variance-maximising—and assess their performance across walking modes.

5.3 Statistical Analysis of Reconstruction Methods

Statistical Overview

Objective: Compare how well three reconstruction algorithms (Anatomical, ℓ_1 , Variance) capture EMG variance and examine any influence of walking mode.

Tests:

- Friedman test – global comparison of median explained variance across the three methods.
- Nemenyi post-hoc test – pairwise method comparison whenever the Friedman test is significant.
- Mann–Whitney U tests – assess whether each method’s performance differs between overground and treadmill walking.

We selected vastus lateralis channels using three methods (anatomical selection, variance-based selection, and ℓ_1 -based selection) and reconstructed full EMG using a shallow neural network trained to map the reduced input to synergy activations, which were then recombined with the NMF weight matrix.

Reconstruction explained variance averaged $72\% \pm 9.8\%$ for anatomical selection (4 channels), $65.2\% \pm 10.2\%$ for variance-based selection (10 channels), and $64.1\% \pm 15.1\%$ for ℓ_1 -based selection (10 channels).

5.3.1 Friedman Test for Reconstruction Methods

Friedman Test for Overall Differences

This Friedman Test compares how well each reconstruction method (Anatomical, L1, Variance) captures EMG variance across participants: we applied non-parametric tests to the explained-variance scores across both walking modes.

Null hypothesis (H_0): The distributions of explained variance are equal for Anatomical, L1, and Variance methods.

Alternative hypothesis (H_1): At least one method differs in median explained variance from the others.

Test statistic:

$$\chi^2(2) = 8.0$$

p-value:

$$p = 0.0183$$

Since $p < 0.05$, we rejected the null hypothesis, as evident from the separation of distributions in Figure 5.6.

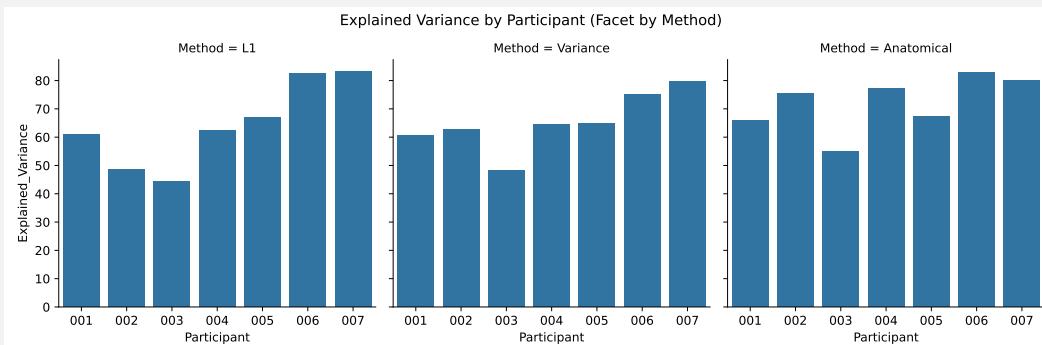


Figure 5.6: Explained-variance scores by participant and reconstruction method.

5.3.2 Nemenyi Post-Hoc Test

Nemenyi Post-Hoc Pairwise Comparisons

This Nemenyi post-hoc test identifies which specific pairs of reconstruction methods differ in explained-variance across methods by comparing their mean-rank differences against a critical threshold.

Null hypothesis (H_0): No difference in mean rank of explained variance between any pair of methods.

Alternative hypothesis (H_1): At least one pair of methods differs in mean rank.

Mean ranks:

- Anatomical: 1.67
- L1: 2.17
- Variance: 2.17

Critical Difference (CD): 1.25

Comparison	Rank diff.	Significant
Anatomical vs. L1	0.50	No
L1 vs. Variance	0.00	No
Anatomical vs. Variance	0.50	No

Table 14: Post-hoc Nemenyi pairwise comparisons of mean rank for explained variance across reconstruction methods.

None of the pairwise rank differences exceed $CD=1.25$, no significant pairwise differences were observed.(Table 14).

5.3.3 Mann–Whitney U Tests

Mann–Whitney U Tests for Mode Effects

This Mann–Whitney U test nonparametrically compares the distributions of explained variance between Overground and Treadmill walking for each reconstruction method, avoiding normality assumptions.

Null hypothesis (H_0): For each method, the distribution of explained variance is

the same in overground versus treadmill walking.

Alternative hypothesis (H_1): For at least one method, explained variance differs between modes.

Note – These Mann–Whitney tests were run on individual strides (92 Overground, 69 Treadmill). Because strides from the same participant are correlated.

Method	U-statistic	p-value	Significant
Anatomical	3	0.317	No
L1	2	0.229	No
Variance	2	0.229	No

Table 15: Mann–Whitney U test results comparing explained variance between overground and treadmill walking for each reconstruction method.

Since all $p > 0.05$, we failed to reject the null hypothesis for any method. Figure 5.7 shows a slight upward shift in the treadmill half-violins, and the joint-plots in Figure 5.8 indicate that most participants scored marginally higher on the treadmill. Nonetheless, these visual trends are not statistically significant, corroborating the tabulated results.

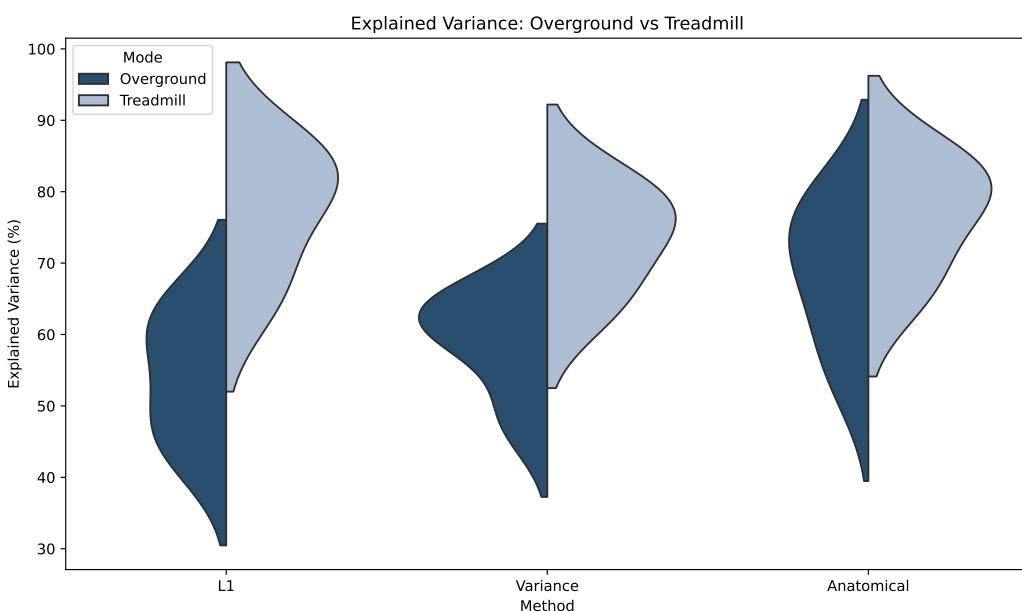


Figure 5.7: Violin distributions of explained-variance scores for each reconstruction method, comparing Overground (left half of each violin) versus Treadmill (right half of each violin) walking.

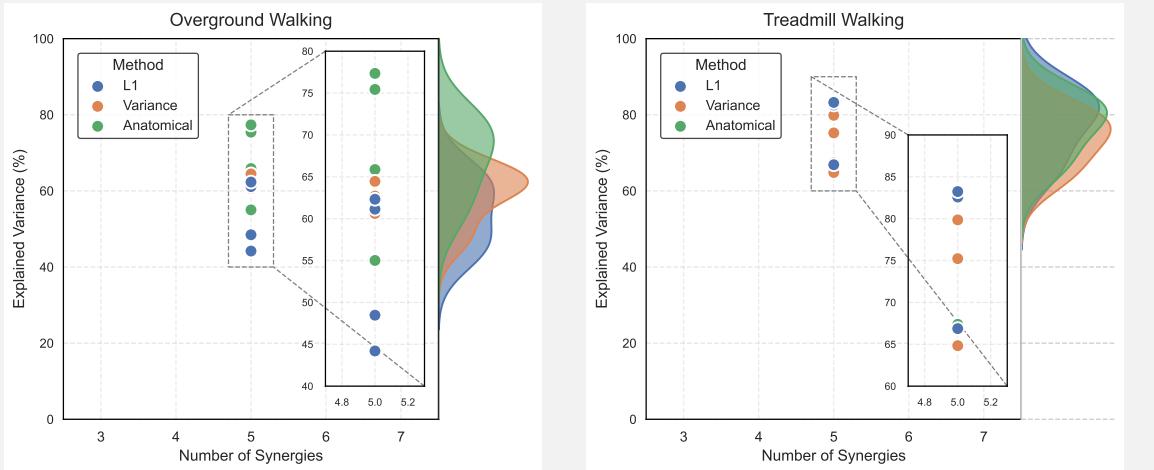


Figure 5.8: Bivariate joint-plots of synergy counts versus explained-variance score for each participant, shown separately for Overground (left) and Treadmill (right) walking.

To demonstrate the practical implications of the statistical findings above, we next evaluated whether a reduced array of vastus lateralis electrodes, combined with a temporally aware CNN–BiLSTM model, can predict tibialis anterior activity with high fidelity. This application tests the extent to which synergy-informed modelling can enable sensor reduction without compromising reconstruction quality.

5.4 Evaluation of Vastus Lateralis→Tibialis Anterior Reconstruction

Analysis Overview

Objective: Evaluate whether a reduced Vastus Lateralis (VL) electrode subset, with and without temporal modelling, can accurately recreate Tibialis Anterior (TA) activity.

Approaches and Metrics:

- Static Pearson correlation – tests instantaneous linear mapping between VL and TA amplitudes.
- CNN–BiLSTM prediction – deep learning model using temporal windows to estimate TA.
- Statistical comparisons:
 - Mann–Whitney U test – compare CNN–BiLSTM correlations between walking modes.
 - One-sided Wilcoxon signed-rank test – test whether CNN–BiLSTM correlations exceed those from the static mapping.
- Visual diagnostics: overlay plots, radar charts, and violin plots illustrate reconstruction quality and consistency.

5.4.1 Static Vastus Lateralis → Tibialis Anterior Correlation Analysis

Static Vastus Lateralis → Tibialis Anterior Pearson Correlations

We computed Pearson correlations between each VL electrode and each TA channel to test whether the reduced VL set linearly spans TA activity.

Participant-wise mean (\pm SD) correlations:

Participant	Mean r	SD r
001	0.068	0.147
002	0.166	0.013
003	0.080	0.024
004	-0.082	0.127
005	-0.048	0.136
006	-0.065	0.026
007	-0.152	0.036

Table 16: Participant-wise mean \pm SD of Pearson r for VL→TA reconstruction by the Static Method

All participants show near-zero (or negative) correlations (group mean $r \approx 0.03 \pm 0.11$; all $|r| < 0.17$), indicating that a simple linear mapping from VL to TA is insufficient.

5.4.2 Convolutional Neural Network–Bidirectional Long Short-Term Memory Temporal Refinement

Convolutional Neural Network–Bidirectional Long Short-Term Memory Tibialis Anterior Reconstruction Correlations

Applying our windowed CNN+BiLSTM model dramatically improved TA reconstruction. For each participant, we computed the mean Pearson r across all TA channels on held-out windows.

Participant-wise mean (\pm SD) correlations:

Participant	Mean r	SD r
001	0.975	0.002
002	0.857	0.013
003	0.983	0.003
004	0.973	0.016
005	0.974	0.008
006	0.917	0.015
007	0.982	0.001

Table 17: Participant-wise mean \pm SD of Pearson r for VL \rightarrow TA reconstruction by the CNN–BiLSTM model.

Group means by condition:

- Overground (001–004): 0.947 ± 0.060
- Treadmill (005–007): 0.958 ± 0.035



Figure 5.9: Radar plots of mean TA reconstruction correlation (\pm SD) for Overground (left) and Treadmill (right) walking.

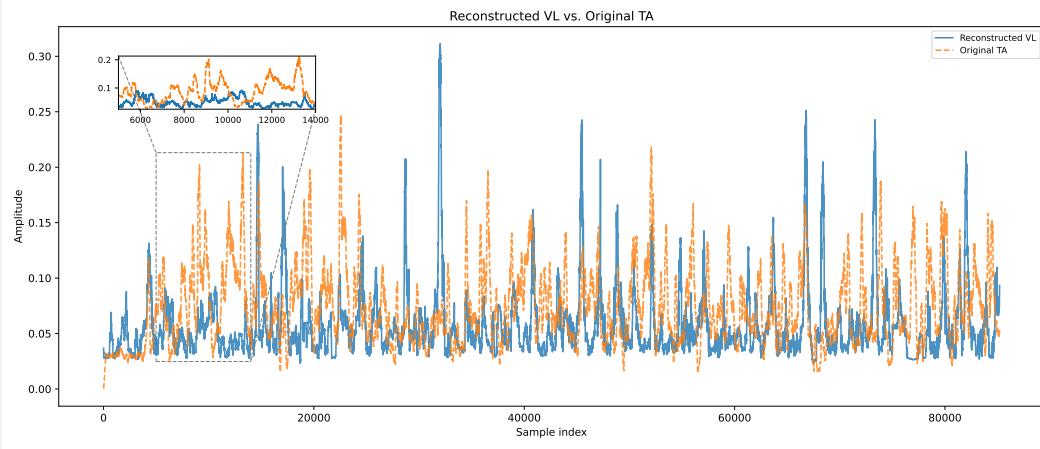


Figure 5.10: CNN–BiLSTM–reconstructed TA signal (dashed orange) overlaid on the original TA signal (solid blue) for a representative participant.

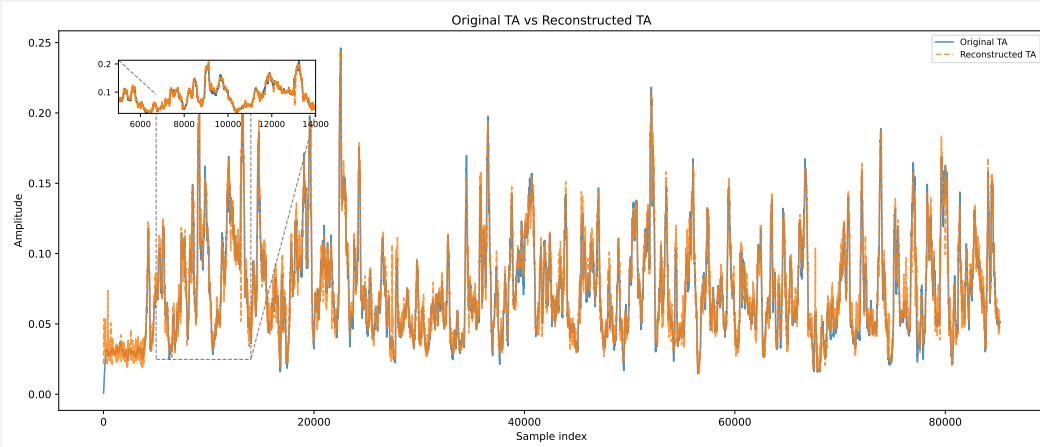


Figure 5.11: CNN–BiLSTM–reconstructed TA signal (dashed orange) overlaid on the original TA signal (solid blue) for a representative participant.

Since the CNN+BiLSTM yields very high correlations ($\approx 0.95\text{--}0.98$), it clearly captures TA dynamics far better than the static VL \rightarrow TA mapping, whose correlations were near-zero in every participant (group mean $r \approx 0.03$, all $|r| < 0.17$).

5.4.3 Walking Mode Comparison of Convolutional Neural Network–Bidirectional Long Short-Term Memory Performance

Mann–Whitney U Test for Mode Effects

This Mann–Whitney U test nonparametrically compares CNN–BiLSTM reconstruction correlations between overground and treadmill walking, without assuming normality.

Null hypothesis (H_0): The distribution of CNN–BiLSTM reconstruction correlations is the same for overground and treadmill walking.

Alternative hypothesis (H_1): The distributions differ between walking conditions.

$$U = 5.0, \quad p = 0.857$$

Since $p > 0.05$, we failed to reject the null hypothesis, consistent with the overlapping radial plots in Figure 5.9, indicating no significant difference in CNN–BiLSTM performance between overground and treadmill walking.

Unit of inference: one mean Pearson r per participant (all 23 strides → single average). Overground = 4 , Treadmill = 3 .

5.4.4 Comparison of Static versus Convolutional Neural Network–Bidirectional Long Short-Term Memory Reconstruction

One-Sided Wilcoxon Signed-Rank Test

This one-sided Wilcoxon signed-rank test assesses whether CNN–BiLSTM correlations exceed static VL→TA correlations across participants.

Null hypothesis (H_0): The median difference in Pearson r between CNN–BiLSTM and static VL→TA correlations is ≤ 0 .

Alternative hypothesis (H_1): The median difference is > 0 (CNN–BiLSTM yields higher correlations).

$$V = 28.0, \quad p = 0.00781$$

Since $p < 0.01$, we rejected the null hypothesis, as reflected in the tight waveform match in Figure 5.11, confirming that the CNN–BiLSTM model produces significantly higher reconstruction correlations than the static linear mapping.

6 Discussion

6.1 Synergy–Phase Alignment Across Walking Modes

Contour plots (Figures 5.1–5.2) and Mann–Whitney U tests with FDR correction (25 comparisons, all adjusted p -values > 0.05 ; Table 12) confirm that the *magnitude* of each synergy’s activation is conserved between overground and treadmill walking. **However**, the **dominant synergy–phase assignments** switch almost completely between conditions (Table 13):

- **Overground:** Synergy 2 → Initial Contact (0–10%), Synergy 5 → Mid-Stance (10–50%), Synergy 3 → Terminal Stance (50–60%), Synergy 4 → Pre-Swing (60–80%), Synergy 1 → Terminal Swing (80–100%).
- **Treadmill:** Synergy 1 → Initial Contact (0–10%), Synergy 4 → Mid-Stance (10–50%), Synergy 2 → Terminal Stance (50–60%), Synergy 3 → Pre-Swing (60–80%), Synergy 5 → Terminal Swing (80–100%).

These systematic flips show that, while each primitive’s temporal profile remains sharply phase-locked, the mapping of which synergy owns each gait subphase is re-scheduled depending on walking context. This pattern implies a dual strategy of conserved timing combined with flexible deployment, allowing centrally stored motor modules to adapt to altered sensory or mechanical constraints.

6.2 Five Muscle Synergies Are Required to Reconstruct Steady-State Electromyographic Signals

Figure 5.3 shows that explained variance increased steeply up to the fourth synergy and gained an additional, statistically significant increment with the fifth. A Friedman statistic of $\chi^2(2) = 14.0$ ($p < 0.001$) confirmed genuine differences among three, four, and five

factors. Our data therefore support a five-synergy solution—consistent with the study by Ivanenko et al. [11].

6.3 Reconstruction Methods: Statistical Equivalence with a Trend Toward Anatomical Specificity

A Friedman test comparing the three reconstruction algorithms (Anatomical, ℓ_1 , Variance) revealed a significant overall effect on explained variance ($\chi^2(2) = 8.0$, $p = 0.0183$). However, Nemenyi post-hoc pairwise comparisons (Table 14) showed that none of the mean-rank differences exceeded the critical difference ($CD = 1.25$), so no individual method pair differed significantly. Moreover, Mann–Whitney U tests assessing mode effects on each method’s performance found no significant differences between overground and treadmill walking (Table 15; all $p > 0.05$). Thus, although the global Friedman result indicates a small but reliable overall difference among methods, the lack of significant pairwise contrasts and mode effects suggests that Anatomical, ℓ_1 , and Variance reconstructions yield practically equivalent variance-explained profiles, allowing choice of algorithm based on interpretability or application context.

6.4 Temporal Convolutional Neural Network–Bidirectional Long Short-Term Memory Model Greatly Improves Vastus Lateralis to Tibialis Anterior Reconstruction

Static VL→TA correlations hovered near zero (Table 16), confirming that instantaneous linear mapping is inadequate. The CNN–BiLSTM, by contrast, achieved mean correlations above 0.95 (Figure 5.9, Table 17). A one-sided Wilcoxon signed-rank test ($p = 0.008$) verified that the temporal model outperformed the static approach. No difference emerged between walking modes ($U = 5$, $p = 0.86$), indicating robust generalisation.

6.5 Physiological and Clinical Implications

The discovery of environment-specific timing shifts cautions against assuming that synergy-derived biomarkers obtained on a treadmill translate directly to overground walking

without temporal adjustment. Nevertheless, the preservation of phase locking implies that control schemes for assistive devices can still leverage synergy timing, provided that phase boundaries are adapted to context. The five-synergy requirement further warns against overly parsimonious models, particularly in pathological gait, where additional or merged synergies are common [8]. Finally, the superiority of the CNN–BiLSTM underscores the value of temporal context in electrode-reduction strategies for wearable EMG systems.

6.6 Study Limitations

Despite the promising outcomes, our study has several limitations that should be addressed in future work:

1. Offline-only Model Evaluation:

All deep learning models used in this study were trained and evaluated on pre-recorded datasets. We have not yet deployed or tested these models in real-time applications. As such, critical metrics such as inference latency, energy efficiency, and robustness to motion artifacts remain uncharacterized. These factors are crucial for future applications in prosthetic or exoskeleton control systems.

2. Limited Detection of Deeper Muscles:

The use of surface EMG inherently restricts our recordings to superficial muscle groups. This means deeper muscles, which likely play important roles in locomotor control and coordination, were not captured in this study. Future work should consider complementary techniques—such as fine-wire EMG or imaging—to gain insights into deeper musculature.

3. Hardware Practicality and Sensor Bulk:

Although the 128-channel HD-sEMG system provided high spatial resolution, it also introduced challenges related to wearability and stability. Two recording sessions failed due to complete probe detachment. While we performed a preliminary electrode-reduction test, the scope was limited to a single muscle pair and did not resolve broader issues of system portability and ease of use.

4. Narrow Scope of Electrode-Reduction Validation:

Our electrode-reduction approach focused solely on reconstructing tibialis anterior activity from vastus lateralis signals. While this serves as an initial demonstration of feasibility, we have not yet validated whether this mapping generalizes to other muscles or supports broader electrode reduction strategies. Future investigations should extend this analysis across multiple muscles and sensor configurations.

7 Conclusion

This thesis set out to uncover the fundamental building blocks of human gait and to translate that insight into practical strategies for wearable assistive systems. Our combined use of high-density surface electromyography, marker-based kinematics, non-negative matrix factorization, and temporal deep learning yields three major advances.

First, we establish that five phase-locked muscle synergies suffice to reconstruct over 92% of lower-limb EMG variance during steady walking. The five synergies occupy distinct portions of the gait cycle—two spanning approximately 10%, two around 20%, and one broader synergy of about 40%—confirming that the nervous system deploys modular activation patterns in a consistent sequence. The statistically significant jump in explained variance from four to five synergies underscores the importance of including a fifth component for accurate and comprehensive EMG decoding.

Second, using contour-plot analysis and two-sided Mann–Whitney U tests with Benjamini–Hochberg FDR correction at $q = 0.05$, we demonstrate that synergy composition and activation magnitude remain invariant across treadmill and overground walking, yet the dominant synergy–phase mapping shifts between modes for all five gait subphases (Table 13). In other words, the spinal pattern generator retains a stable library of modules but flexibly reschedules their phase boundaries in response to altered sensory feedback. This finding refines the conceptual model of modular locomotor control and underscores the need for context-aware phase definitions in translational applications.

Third, we show that temporal deep learning (CNN–BiLSTM) dramatically outperforms static linear mapping for electrode-reduction tasks. Whereas instantaneous VL→TA correlation was negligible, our windowed model achieved correlations above 0.95 and generalized robustly to different walking modes. This result points to the critical role of temporal context in sparse-sensor reconstruction and lays the foundation for energy-efficient, low-channel-count EMG systems.

Taken together, these contributions advance both theoretical understanding and algorithmic design for human locomotion. The five-synergy framework offers a quantitative benchmark for healthy and pathological gait, while the modular, time-shifted activation pattern informs the development of intelligent prostheses and exoskeleton controllers. Moreover, integrating synergy-based features with real-time deep-learning pipelines makes online phase detection tractable on embedded platforms.

Future work will translate this pipeline into wearable hardware to evaluate inference latency, power consumption, and motion-artifact resilience. It will also extend validation to clinical populations—characterized by synergy merging or fractionation—and incorporate invasive or imaging modalities to capture deeper musculature. Ultimately, this synergy-aware framework aims to deliver adaptive, lightweight assistive devices that seamlessly support users across diverse locomotor environments.

8 Appendix

Best BiLSTM Model Configuration

Hyperparameter	Description	Best Value
<code>units_1</code>	Number of units in first LSTM layer	32
<code>bidirectional</code>	Use bidirectional LSTM	True
<code>dropout_1</code>	Dropout rate after first LSTM layer	0.2
<code>second_lstm</code>	Flag to add a second LSTM layer	False
<code>units_2</code>	Units in second LSTM layer (if used)	—
<code>dropout_2</code>	Dropout rate after second LSTM layer	—
<code>fc_units</code>	Number of units in the Dense layer	32
<code>fc_dropout</code>	Dropout rate after Dense layer	0.3
<code>learning_rate</code>	Learning rate for Adam optimizer	8.05×10^{-4}

Table 18: Best hyperparameter configuration of the BiLSTM-based model, selected across all participants. The configuration includes hidden size, dropout settings, and optimizer parameters. Optional second LSTM layer was disabled during final training.

Best CNN Model Configuration

Hyperparameter	Description	Best Value
<code>conv_layers</code>	Number of convolutional layers	1
<code>filters_i</code>	Filters in the first Conv1D layer	64
<code>kernel_size_i</code>	Kernel size for first Conv1D layer	3
<code>activation_i</code>	Activation function	ReLU
<code>dropout_i</code>	Apply dropout after Conv1D	True
<code>dropout_rate_i</code>	Dropout rate after Conv1D layer	0.2
<code>dense_units</code>	Units in the fully connected layer	128
<code>dense_dropout</code>	Dropout rate after dense layer	0.2

Table 19: Best hyperparameter configuration of the CNN-based model, optimized across all participants. This includes convolutional depth, kernel parameters, dropout rates, and fully connected layer settings.

Best Transformer Model Configuration

Hyperparameter	Description	Best Value
<code>embedding_dim</code>	Dimension of input embeddings	64
<code>kernel_size</code>	Kernel size for convolutional encoder	5
<code>num_blocks</code>	Number of Transformer blocks	6
<code>num_heads</code>	Number of attention heads	4
<code>ff_dim</code>	Feed-forward layer dimension	256
<code>block_dropout</code>	Dropout rate after each block	0.1
<code>pooling_type</code>	Pooling operation before output	Mean
<code>head_dropout</code>	Dropout rate after attention head	0.3
<code>learning_rate</code>	Learning rate for Adam optimizer	2.456×10^{-5}

Table 20: Best hyperparameter configuration of the Transformer-based model, determined across all participants. Includes embedding dimensions, number of attention heads, feed-forward network size, dropout rates, and optimizer learning rate.

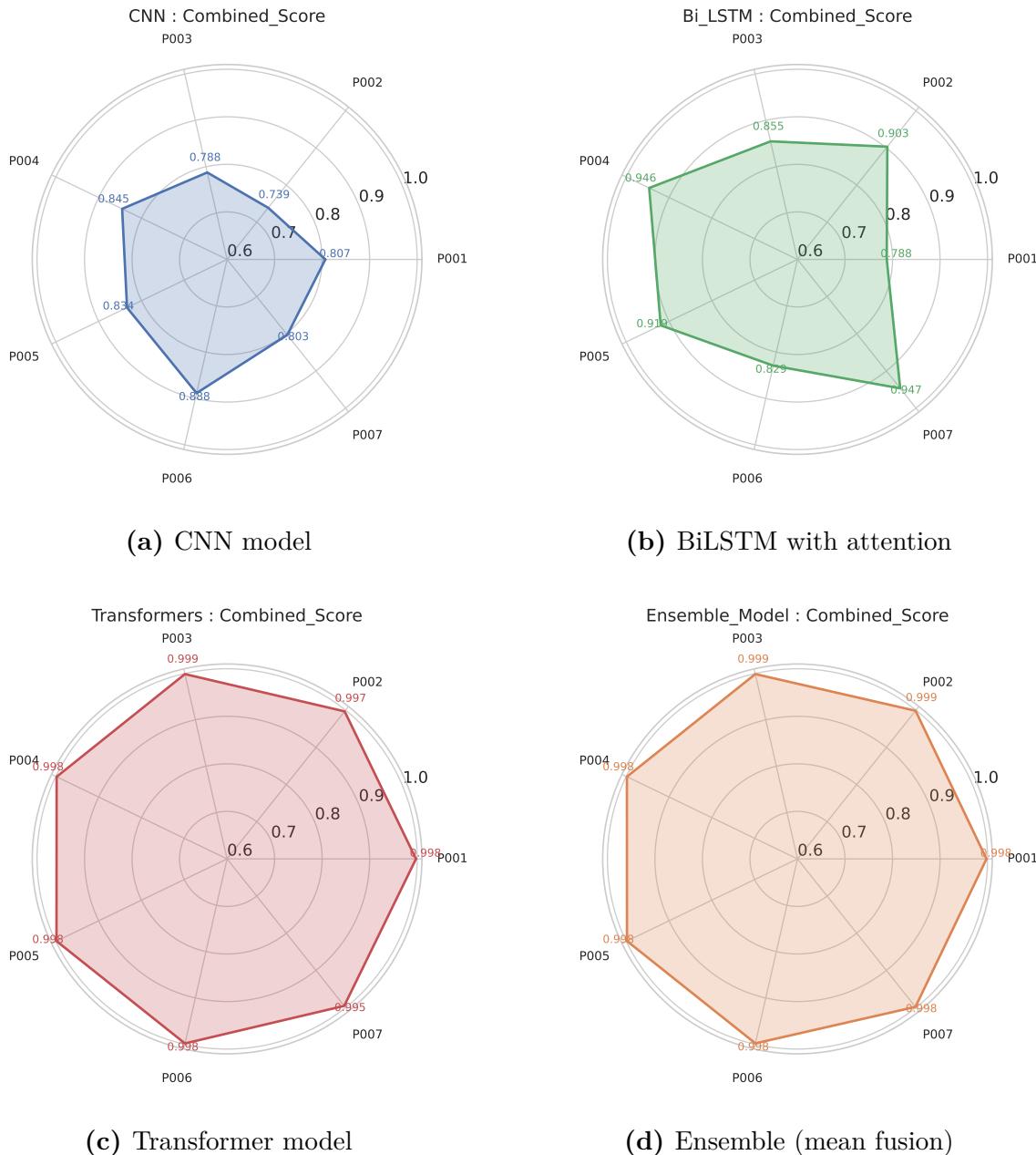


Figure 8.1: Summary of model-wise participant evaluation. Subfigures (a–d) display radar plots of the combined score—calculated as the mean of weighted F1 score and weighted ROC AUC—for each participant (P001–P007) across four architectures: CNN, BiLSTM, Transformer, and the Ensemble model.

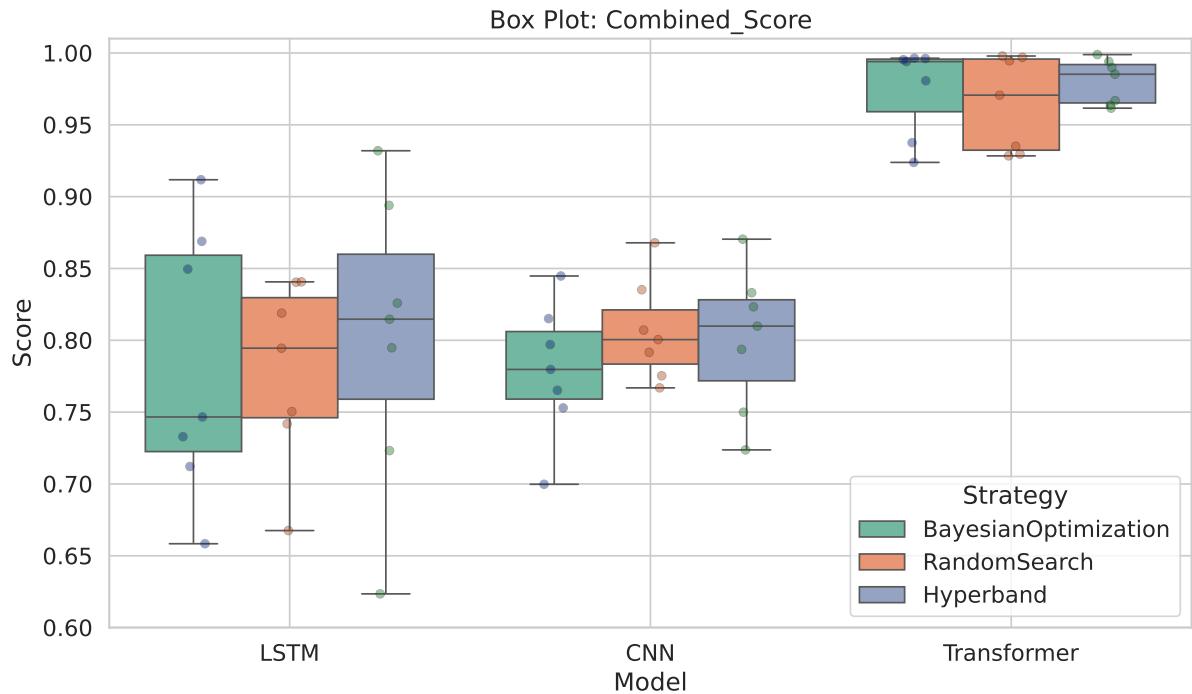


Figure 8.2: Distribution of combined validation scores, defined as the mean of the weighted F1 score and weighted ROC AUC (Equation 4.52), for three model architectures—BiLSTM, CNN, and Transformer—across seven participants. Each model was optimized via Bayesian Optimization, Random Search, and Hyperband. The Transformer consistently achieved the highest average performance and the lowest inter-run variability.

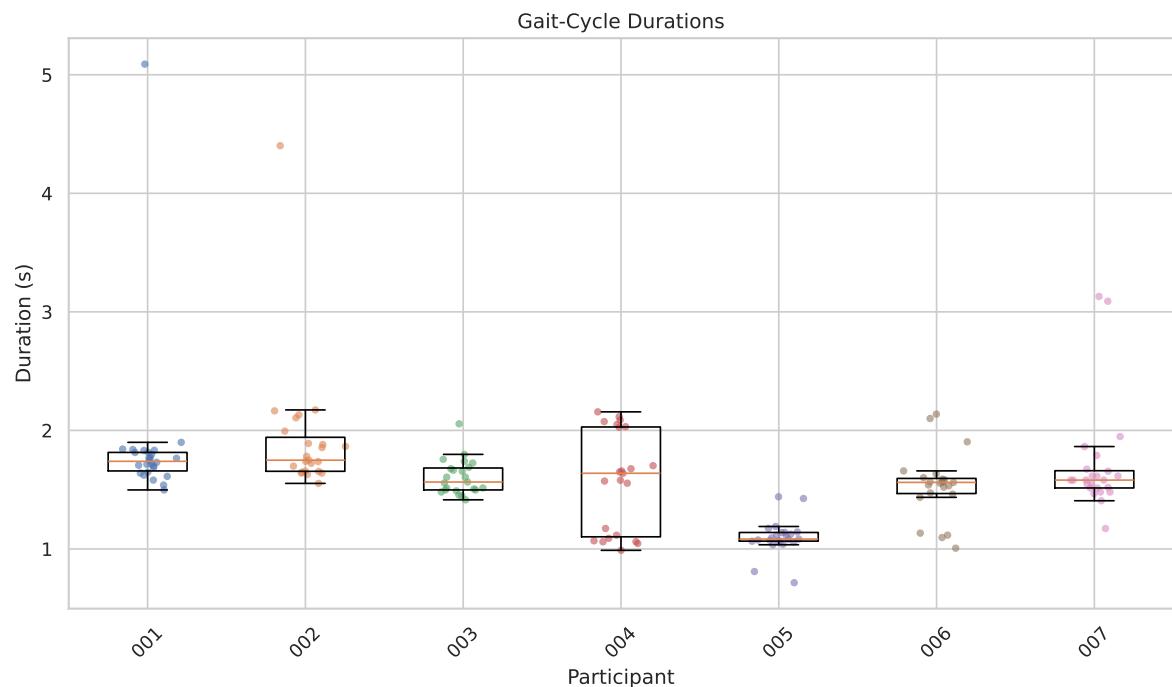


Figure 8.3: Participant-wise variability in gait cycle durations, shown as boxplots with swarm overlays. Despite individual differences, the majority of cycles lie between 1.5s and 2.0s.

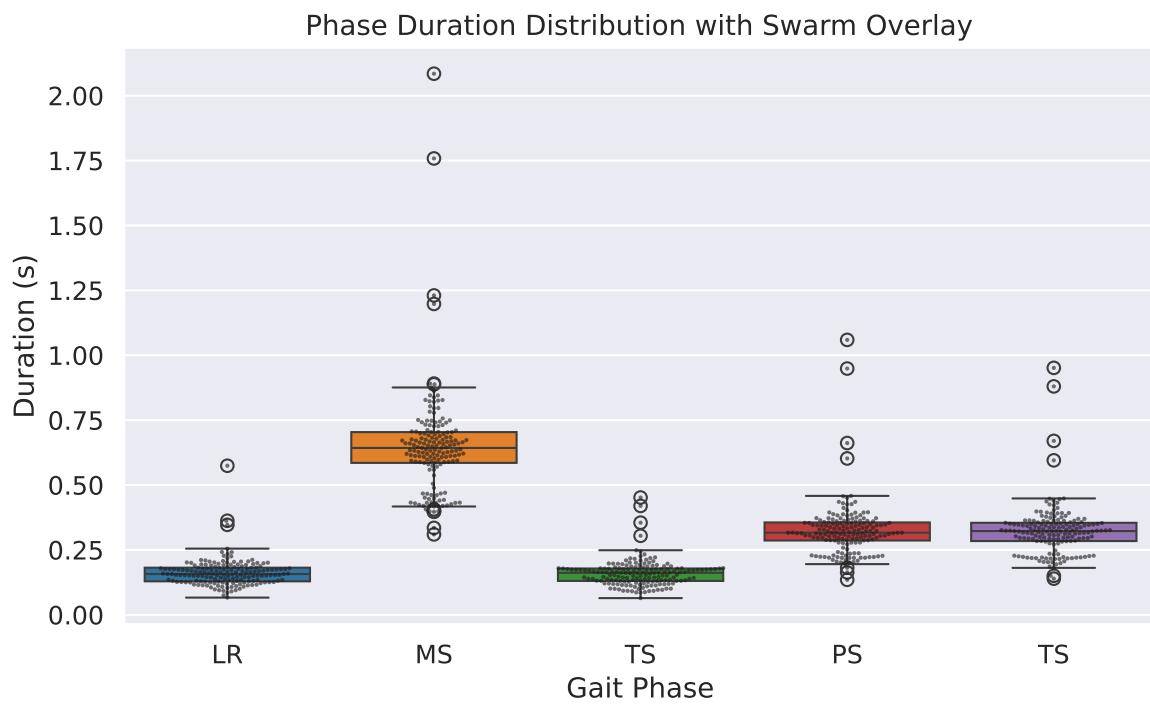


Figure 8.4: Phase-wise duration distributions for all detected gait phases, with swarm points indicating individual stride events. Boxplots correspond to: LR = Loading Response, MS = Midstance, TS = Terminal Stance, PS = Preswing, and TS = Terminal Swing.

Bibliography

- [1] J. Taborri, E. Palermo, S. Rossi, and P. Cappa. “Gait Partitioning Methods: A Systematic Review”. en. In: *Sensors* 16.1 (2016). Number: 1 Publisher: Multi-disciplinary Digital Publishing Institute, p. 66. DOI: [10.3390/s16010066](https://doi.org/10.3390/s16010066). URL: <https://www.mdpi.com/1424-8220/16/1/66> (cit. on p. 1).
- [2] R. Di Gregorio and L. Vocenas. “Identification of Gait-Cycle Phases for Prostheses Control”. en. In: *Biomimetics* 6.2 (2021). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 22. DOI: [10.3390/biomimetics6020022](https://doi.org/10.3390/biomimetics6020022). URL: <https://www.mdpi.com/2313-7673/6/2/22> (cit. on p. 1).
- [3] H. Park, S. Han, J. Sung, S. Hwang, I. Youn, and S.-J. Kim. “Classification of gait phases based on a machine learning approach using muscle synergy”. English. In: *Frontiers in Human Neuroscience* 17 (2023). Publisher: Frontiers. DOI: [10.3389/fnhum.2023.1201935](https://doi.org/10.3389/fnhum.2023.1201935). URL: <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2023.1201935/full> (cit. on pp. 1, 36).
- [4] S. Hagio, M. Fukuda, and M. Kouzaki. “Identification of muscle synergies associated with gait transition in humans”. English. In: *Frontiers in Human Neuroscience* 9 (2015). Publisher: Frontiers. DOI: [10.3389/fnhum.2015.00048](https://doi.org/10.3389/fnhum.2015.00048). URL: <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2015.00048/full> (cit. on p. 1).
- [5] L. H. Ting and J. M. Macpherson. “A Limited Set of Muscle Synergies for Force Control During a Postural Task”. In: *Journal of Neurophysiology* 93.1 (2005). Publisher: American Physiological Society, pp. 609–613. DOI: [10.1152/jn.00681.2004](https://doi.org/10.1152/jn.00681.2004). URL: <https://journals.physiology.org/doi/full/10.1152/jn.00681.2004> (cit. on p. 1).
- [6] J. Roh, W. Z. Rymer, and R. F. Beer. “Robustness of muscle synergies underlying three-dimensional force generation at the hand in healthy humans”. In: *Journal of Neurophysiology* 107.8 (2012). Publisher: American Physiological Society, pp. 2123–2142. DOI: [10.1152/jn.00173.2011](https://doi.org/10.1152/jn.00173.2011). URL: <https://journals.physiology.org/doi/full/10.1152/jn.00173.2011> (cit. on p. 1).

- [7] S. Hagio and M. Kouzaki. “The flexible recruitment of muscle synergies depends on the required force-generating capability”. In: *Journal of Neurophysiology* 112.2 (2014). Publisher: American Physiological Society, pp. 316–327. DOI: [10.1152/jn.00109.2014](https://doi.org/10.1152/jn.00109.2014). URL: <https://journals.physiology.org/doi/full/10.1152/jn.00109.2014> (cit. on p. 1).
- [8] D. J. Clark, L. H. Ting, F. E. Zajac, R. R. Neptune, and S. A. Kautz. “Merging of Healthy Motor Modules Predicts Reduced Locomotor Performance and Muscle Coordination Complexity Post-Stroke”. In: *Journal of Neurophysiology* 103.2 (2010). Publisher: American Physiological Society, pp. 844–857. DOI: [10.1152/jn.00825.2009](https://doi.org/10.1152/jn.00825.2009). URL: <https://journals.physiology.org/doi/full/10.1152/jn.00825.2009> (cit. on pp. 1, 14, 26, 81).
- [9] S. A. Chvatal and L. H. Ting. “Common muscle synergies for balance and walking”. English. In: *Frontiers in Computational Neuroscience* 7 (2013). Publisher: Frontiers. DOI: [10.3389/fncom.2013.00048](https://doi.org/10.3389/fncom.2013.00048). URL: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2013.00048/full> (cit. on p. 1).
- [10] D. Borzelli, C. De Marchis, A. Quercia, P. De Pasquale, A. Casile, A. Quartarone, R. S. Calabro, and A. d’Avella. “Muscle Synergy Analysis as a Tool for Assessing the Effectiveness of Gait Rehabilitation Therapies: A Methodological Review and Perspective”. en. In: *Bioengineering* 11.8 (2024). Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 793. DOI: [10.3390/bioengineering11080793](https://doi.org/10.3390/bioengineering11080793). URL: <https://www.mdpi.com/2306-5354/11/8/793> (cit. on p. 1).
- [11] Y. P. Ivanenko, R. E. Poppele, and F. Lacquaniti. “Five basic muscle activation patterns account for muscle activity during human locomotion”. In: *The Journal of Physiology* 556.Pt 1 (2004), pp. 267–282. DOI: [10.1113/jphysiol.2003.057174](https://doi.org/10.1113/jphysiol.2003.057174). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1664897/> (cit. on pp. 2, 80).
- [12] S. A. Safavynia, G. Torres-Oviedo, and L. H. Ting. “Muscle Synergies: Implications for Clinical Evaluation and Rehabilitation of Movement”. In: *Topics in spinal cord injury rehabilitation* 17.1 (2011), pp. 16–24. DOI: [10.1310/sci1701-16](https://doi.org/10.1310/sci1701-16). URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3143193/> (cit. on p. 2).
- [13] N. Roth, A. Küderle, M. Ullrich, T. Gladow, F. Marxreiter, J. Klucken, B. M. Eskofier, and F. Kluge. “Hidden Markov Model based stride segmentation on unsupervised free-living gait data in Parkinson’s disease patients”. In: *Journal of NeuroEngineering and Rehabilitation* 18.1 (2021), p. 93. DOI: [10.1186/s12984-021-00883-7](https://doi.org/10.1186/s12984-021-00883-7). URL: <https://doi.org/10.1186/s12984-021-00883-7> (cit. on pp. 4, 37).

- [14] R. A. Bogey, J. Perry, E. L. Bontrager, and J. K. Gronley. “Comparison of across-subject EMG profiles using surface and multiple indwelling wire electrodes during gait”. In: *Journal of Electromyography and Kinesiology* 10.4 (2000), pp. 255–259. DOI: [10.1016/S1050-6411\(00\)00015-8](https://doi.org/10.1016/S1050-6411(00)00015-8). URL: <https://www.sciencedirect.com/science/article/pii/S1050641100000158> (cit. on p. 5).
- [15] D. Ao, M. S. Shourijeh, C. Patten, and B. J. Fregly. “Evaluation of Synergy Extrapolation for Predicting Unmeasured Muscle Excitations from Measured Muscle Synergies”. English. In: *Frontiers in Computational Neuroscience* 14 (2020). Publisher: Frontiers. DOI: [10.3389/fncom.2020.588943](https://doi.org/10.3389/fncom.2020.588943). URL: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2020.588943/full> (cit. on p. 5).
- [16] J. Betts. *Anatomy and Physiology*. Open Textbook Library. OpenStax College, Rice University, 2013. URL: <https://books.google.de/books?id=dvVgngEACAAJ> (cit. on p. 6).
- [17] C. Gopalan. *Essentials of Physiology for Nurse Anesthetists I*. 2025. URL: [https://med.libretexts.org/Courses/Southern_Illinois_University_Edwardsville/Essentials_of_Physiology_for_Nurse_Anesthetists_I_\(Gopalan\)](https://med.libretexts.org/Courses/Southern_Illinois_University_Edwardsville/Essentials_of_Physiology_for_Nurse_Anesthetists_I_(Gopalan)) (cit. on p. 6).
- [18] S. López. “Design and Construction of an EMG Multichannel Acquisition System Prototype”. In: 2012. URL: <https://www.semanticscholar.org/paper/Design-and-Construction-of-an-EMG-Multichannel-L%C3%B3pez/3b8de6257ad1f59c85af4a791e69cd4f0c0888ad> (cit. on pp. 6–8, 10).
- [19] H. Huxley and J. Hanson. “Changes in the Cross-Striations of Muscle during Contraction and Stretch and their Structural Interpretation”. en. In: *Nature* 173.4412 (1954). Publisher: Nature Publishing Group, pp. 973–976. DOI: [10.1038/173973a0](https://doi.org/10.1038/173973a0). URL: <https://www.nature.com/articles/173973a0> (cit. on pp. 7, 8).
- [20] D. E. Rassier, B. R. MacIntosh, and W. Herzog. “Length dependence of active force production in skeletal muscle”. In: *Journal of Applied Physiology* 86.5 (1999). Publisher: American Physiological Society, pp. 1445–1457. DOI: [10.1152/jappl.1999.86.5.1445](https://doi.org/10.1152/jappl.1999.86.5.1445). URL: <https://jappl.physiology.org/doi/full/10.1152/jappl.1999.86.5.1445> (cit. on p. 8).
- [21] A. F. Huxley and R. Niedergerke. “Structural Changes in Muscle During Contraction: Interference Microscopy of Living Muscle Fibres”. en. In: *Nature* 173.4412 (1954). Publisher: Nature Publishing Group, pp. 971–973. DOI: [10.1038/173971a0](https://doi.org/10.1038/173971a0). URL: <https://www.nature.com/articles/173971a0> (cit. on p. 8).

- [22] V. Cnejevici. *Comparative Analysis and Optimization of AI Models for Hand Movement Estimation Using High-Density Surface Electromyography [unpublished]*. Master's thesis. Unpublished master's thesis (cit. on pp. 9, 12, 13, 31).
- [23] G. Kamen and D. Gabriel. *Essentials of Electromyography*. Human Kinetics, 2010. URL: <https://books.google.de/books?id=k--6vgEACAAJ> (cit. on pp. 10, 11).
- [24] M. Hammelsbeck and W. Rathmayer. "Intracellular Na+, K+ and Cl activity in tonic and phasic muscle fibers of the crab *Eriphia*". en. In: *Pflügers Archiv* 413.5 (1989), pp. 487–492. DOI: [10.1007/BF00594178](https://doi.org/10.1007/BF00594178). URL: <https://doi.org/10.1007/BF00594178> (cit. on p. 10).
- [25] K. Takakusaki. "Neurophysiology of gait: From the spinal cord to the frontal lobe". en. In: *Movement Disorders* 28.11 (2013), pp. 1483–1491. DOI: [10.1002/mds.25669](https://doi.org/10.1002/mds.25669). URL: <https://movementdisorders.onlinelibrary.wiley.com/doi/10.1002/mds.25669> (cit. on p. 13).
- [26] J. P. McCabe, S. Pundik, and J. J. Daly. "Targeting CNS Neural Mechanisms of Gait in Stroke Neurorehabilitation". eng. In: *Brain Sciences* 12.8 (2022), p. 1055. DOI: [10.3390/brainsci12081055](https://doi.org/10.3390/brainsci12081055) (cit. on p. 13).
- [27] R. Ranganathan and C. Krishnan. "Extracting synergies in gait: using EMG variability to evaluate control strategies". eng. In: *Journal of Neurophysiology* 108.5 (2012), pp. 1537–1544. DOI: [10.1152/jn.01112.2011](https://doi.org/10.1152/jn.01112.2011) (cit. on p. 14).
- [28] D. D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". en. In: *Nature* 401.6755 (1999). Publisher: Nature Publishing Group, pp. 788–791. DOI: [10.1038/44565](https://doi.org/10.1038/44565). URL: <https://www.nature.com/articles/44565> (cit. on pp. 14, 26).
- [29] M. C. Tresch, V. C. K. Cheung, and A. d'Avella. "Matrix factorization algorithms for the identification of muscle synergies: evaluation on simulated and experimental data sets". eng. In: *Journal of Neurophysiology* 95.4 (2006), pp. 2199–2212. DOI: [10.1152/jn.00222.2005](https://doi.org/10.1152/jn.00222.2005) (cit. on pp. 14, 26).
- [30] S. Israely, G. Leisman, C. C. Machluf, and E. Carmeli. "Muscle Synergies Control during Hand-Reaching Tasks in Multiple Directions Post-stroke". English. In: *Frontiers in Computational Neuroscience* 12 (2018). Publisher: Frontiers. DOI: [10.3389/fncom.2018.00010](https://doi.org/10.3389/fncom.2018.00010). URL: <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2018.00010/full> (cit. on p. 14).

- [31] A. Kopelevich. *Adapted Gait Cycle.jpg*. en. URL: https://www.physio-pedia.com/File:Adapted_Gait_Cycle.jpg (cit. on p. 15).
- [32] D. J. Magee and R. C. Manske. *Orthopedic Physical Assessment*. en. Google-Books-ID: YdvnuQEACAAJ. Elsevier, 2020 (cit. on pp. 16–18).
- [33] J. B. Webster and D. P. Murphy. *Atlas of Orthoses and Assistive Devices E-Book*. en. Google-Books-ID: 6RNBDwAAQBAJ. Elsevier Health Sciences, 2017 (cit. on pp. 16–18).
- [34] A. Hazari, A. G. Maiya, and T. V. Nagda. “Conceptual Biomechanics and Kinesiology”. en. In: *Springer eBooks* (2021). DOI: [10.1007/978-981-16-4991-2](https://doi.org/10.1007/978-981-16-4991-2). URL: https://www.academia.edu/108120163/Conceptual_Biomechanics_and_Kinesiology (cit. on pp. 16–18).
- [35] *Kinesiology of the Musculoskeletal System*. en-US. 2009. URL: <https://shop.elsevier.com/books/kinesiology-of-the-musculoskeletal-system/neumann/978-0-323-03989-5> (cit. on p. 17).
- [36] K. Hornik, M. Stinchcombe, and H. White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989). Place: Netherlands Publisher: Elsevier Science, pp. 359–366. DOI: [10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8) (cit. on p. 18).
- [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. arXiv:1706.03762 [cs]. 2023. DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762). URL: <http://arxiv.org/abs/1706.03762> (cit. on pp. 19, 42, 47, 49, 55).
- [38] A. Leardini, Z. Sawacha, G. Paolini, S. Ingrosso, R. Nativo, and M. G. Benedetti. “A new anatomically based protocol for gait analysis in children”. en. In: *Gait & Posture* 26.4 (2007), pp. 560–571. DOI: [10.1016/j.gaitpost.2006.12.018](https://doi.org/10.1016/j.gaitpost.2006.12.018). URL: <https://linkinghub.elsevier.com/retrieve/pii/S096636207000124> (cit. on pp. 21, 22).
- [39] A. Hyvärinen. “Fast and robust fixed-point algorithms for independent component analysis”. en. In: *IEEE Transactions on Neural Networks* 10.3 (1999), pp. 626–634. DOI: [10.1109/72.761722](https://doi.org/10.1109/72.761722). URL: <http://ieeexplore.ieee.org/document/761722/> (cit. on p. 23).
- [40] J. Kirsten. *Optimizing Hand Kinematic Predictions through Single and Combinatorial Feature Analysis of Electromyography Signals [unpublished]*. English. Master’s thesis. Unpublished master’s thesis (cit. on p. 25).

- [41] S. Yağmur Günay, F. Quivira, and D. Erdoğmuş. “Muscle Synergy-based Grasp Classification for Robotic Hand Prosthetics”. eng. In: *The ... International Conference on PErvasive Technologies Related to Assistive Environments: PETRA ... International Conference on PErvasive Technologies Related to Assistive Environments* 2017 (2017), pp. 335–338. DOI: [10.1145/3056540.3076208](https://doi.org/10.1145/3056540.3076208) (cit. on p. 25).
- [42] A. d’Avella, P. Saltiel, and E. Bizzi. “Combinations of muscle synergies in the construction of a natural motor behavior”. en. In: *Nature Neuroscience* 6.3 (2003). Publisher: Nature Publishing Group, pp. 300–308. DOI: [10.1038/nn1010](https://doi.org/10.1038/nn1010). URL: <https://www.nature.com/articles/nn1010> (cit. on p. 26).
- [43] D. Lee and H. Seung. “Algorithms for Non-negative Matrix Factorization”. In: *Adv. Neural Inform. Process. Syst.* 13 (2001) (cit. on p. 26).
- [44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. URL: https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html (cit. on pp. 26, 47).
- [45] P. et al. *ReduceLROnPlateau — PyTorch 2.7 documentation*. URL: https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html (cit. on p. 28).
- [46] D. A. Winter. *Biomechanics and motor control of human movement*. eng. 4. ed. OCLC: 699103830. Hoboken, NJ: Wiley, 2009. URL: <http://digitool.hbz-nrw.de:1801/webclient/DeliveryManager?pid=3309881> (cit. on pp. 32, 34).
- [47] J. Hamill, K. M. Knutzen, and T. R. Derrick. *Biomechanical Basis of Human Movement*. 4th ed. Philadelphia, PA: Lippincott Williams & Wilkins, 2014 (cit. on p. 33).
- [48] R. A. Serway and J. W. Jewett. *Physics for Scientists and Engineers with Modern Physics*. 10th ed. Boston, MA: Cengage Learning, 2018 (cit. on pp. 33, 34).
- [49] J. G. Richards. “The measurement of human motion: A comparison of commercially available systems”. In: *Human Movement Science* 18.5 (1999), pp. 589–602. DOI: [10.1016/S0167-9457\(99\)00023-8](https://doi.org/10.1016/S0167-9457(99)00023-8). URL: <https://www.sciencedirect.com/science/article/pii/S0167945799000238> (cit. on p. 34).

- [50] G. Giakas and V. Baltzopoulos. “A comparison of automatic filtering techniques applied to biomechanical walking data”. In: *Journal of Biomechanics* 30.8 (1997), pp. 847–850. DOI: [10.1016/S0021-9290\(97\)00042-0](https://doi.org/10.1016/S0021-9290(97)00042-0). URL: <https://www.sciencedirect.com/science/article/pii/S0021929097000420> (cit. on p. 34).
- [51] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analysis*. 5th ed. Hoboken, NJ: Wiley, 2012 (cit. on p. 35).
- [52] V. Agostini, M. Ghislieri, S. Rosati, G. Balestra, G. Dotti, and M. Knaflitz. “Statistical Gait Analysis Based on Surface Electromyography”. en. In: *Medicine-Based Informatics and Engineering*. Ed. by F. Simini and P. Bertemes-Filho. Cham: Springer International Publishing, 2022, pp. 23–35. DOI: [10.1007/978-3-030-87845-0_2](https://doi.org/10.1007/978-3-030-87845-0_2). URL: https://doi.org/10.1007/978-3-030-87845-0_2 (cit. on p. 36).
- [53] B. M. V. Guerra, M. Schmid, S. Sozzi, S. Pizzocaro, A. M. De Nunzio, and S. Ramat. “A Recurrent Deep Network for Gait Phase Identification from EMG Signals During Exoskeleton-Assisted Walking”. en. In: *Sensors* 24.20 (2024). Number: 20 Publisher: Multidisciplinary Digital Publishing Institute, p. 6666. DOI: [10.3390/s24206666](https://doi.org/10.3390/s24206666). URL: <https://www.mdpi.com/1424-8220/24/20/6666> (cit. on p. 36).
- [54] L. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989). Conference Name: Proceedings of the IEEE, pp. 257–286. DOI: [10.1109/5.18626](https://doi.org/10.1109/5.18626). URL: <https://ieeexplore.ieee.org/document/18626/?arnumber=18626> (cit. on p. 36).
- [55] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*. arXiv:1603.06560 [cs]. 2018. DOI: [10.48550/arXiv.1603.06560](https://doi.org/10.48550/arXiv.1603.06560). URL: <http://arxiv.org/abs/1603.06560> (cit. on pp. 37, 55).
- [56] L. Li, K. G. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar. *A System for Massively Parallel Hyperparameter Tuning*. arXiv preprint arXiv:1810.05934. Presented at the Conference on Machine Learning and Systems (MLSys) 2020. 2018. DOI: [10.48550/arXiv.1810.05934](https://doi.org/10.48550/arXiv.1810.05934). URL: <https://arxiv.org/abs/1810.05934> (cit. on p. 37).
- [57] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Comput.* 9.8 (1997), pp. 1735–1780. 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> (cit. on pp. 39, 47).
- [58] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. 1st ed. Adaptive Computation and Machine Learning. Cambridge, MA: The MIT Press, 2016. URL:

<https://mitpress.mit.edu/9780262035613/deep-learning/> (cit. on pp. 39, 41, 55).

- [59] H. Zhan, J. Kou, Y. Cao, Q. Guo, J. Zhang, and Y. Shi. “Human Gait phases recognition based on multi-source data fusion and BiLSTM attention neural network”. en. In: *Measurement* 238 (2024), p. 115396. DOI: [10.1016/j.measurement.2024.115396](https://doi.org/10.1016/j.measurement.2024.115396). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0263224124012818> (cit. on pp. 39, 41, 42).
- [60] M. Schuster and K. Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing* 45.11 (1997). Conference Name: IEEE Transactions on Signal Processing, pp. 2673–2681. DOI: [10.1109/78.650093](https://doi.org/10.1109/78.650093). URL: <https://ieeexplore.ieee.org/document/650093> (cit. on p. 41).
- [61] A. Graves and J. Schmidhuber. “Framewise phoneme classification with bidirectional LSTM and other neural network architectures”. en. In: *Neural Networks* 18.5-6 (2005), pp. 602–610. DOI: [10.1016/j.neunet.2005.06.042](https://doi.org/10.1016/j.neunet.2005.06.042). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0893608005001206> (cit. on p. 41).
- [62] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998). Conference Name: Proceedings of the IEEE, pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791). URL: <https://ieeexplore.ieee.org/document/726791> (cit. on p. 44).
- [63] J. van den Hoogen. “Time Series Analysis Using Convolutional Neural Networks”. Doctoral Thesis. Tilburg, Netherlands: Tilburg University, 2025. URL: https://pure.uvt.nl/ws/portalfiles/portal/110150436/van_den_Hoogen_Time_14-01-2025_HR_.pdf (cit. on pp. 44, 45).
- [64] H. Alaskar. “Deep Learning of EMG Time–Frequency Representations for Identifying Normal and Aggressive Actions”. In: *International Journal of Computer Science and Network Security* 18.12 (2018), pp. 16–25 (cit. on p. 44).
- [65] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. “Convolutional Sequence to Sequence Learning”. en. In: *Proceedings of the 34th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, 2017, pp. 1243–1252. URL: <https://proceedings.mlr.press/v70/gehring17a.html> (cit. on p. 44).
- [66] A. Jacovi, O. Sar Shalom, and Y. Goldberg. “Understanding Convolutional Neural Networks for Text Classification”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Ed. by T. Linzen, G. Chrupała, and A. Alishahi. Brussels, Belgium: Association for Compu-

tational Linguistics, 2018, pp. 56–65. DOI: [10.18653/v1/W18-5408](https://doi.org/10.18653/v1/W18-5408). URL: <https://aclanthology.org/W18-5408/> (cit. on p. 45).

- [67] A. Javidani and A. Mahmoudi-Aznaveh. “Learning Representative Temporal Features for Action Recognition”. In: *Multimedia Tools and Applications* 81.3 (2022). arXiv:1802.06724 [cs], pp. 3145–3163. DOI: [10.1007/s11042-021-11022-8](https://doi.org/10.1007/s11042-021-11022-8). URL: <http://arxiv.org/abs/1802.06724> (cit. on p. 45).
- [68] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989). Conference Name: Neural Computation, pp. 541–551. DOI: [10.1162/neco.1989.1.4.541](https://doi.org/10.1162/neco.1989.1.4.541). URL: <https://ieeexplore.ieee.org/document/6795724> (cit. on p. 47).
- [69] D. Carneros-Prado, S. González-Velázquez, C. C. Dobrescu, I. González, J. Fontecha, and R. Hervás. “Automation of observational gait assessment through an optical 3D motion system and transformers”. en. In: *Applied Intelligence* 55.4 (2025), p. 250. DOI: [10.1007/s10489-024-06163-w](https://doi.org/10.1007/s10489-024-06163-w). URL: <https://doi.org/10.1007/s10489-024-06163-w> (cit. on pp. 47–49).
- [70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html> (cit. on p. 50).
- [71] S. Ioffe and C. Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv:1502.03167 [cs]. 2015. DOI: [10.48550/arXiv.1502.03167](https://doi.org/10.48550/arXiv.1502.03167). URL: <http://arxiv.org/abs/1502.03167> (cit. on p. 50).
- [72] J. Snoek, H. Larochelle, and R. P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms*. arXiv:1206.2944 [stat]. 2012. DOI: [10.48550/arXiv.1206.2944](https://doi.org/10.48550/arXiv.1206.2944). URL: <http://arxiv.org/abs/1206.2944> (cit. on p. 54).
- [73] J. Bergstra and Y. Bengio. “Random search for hyper-parameter optimization”. In: *J. Mach. Learn. Res.* 13.null (2012), pp. 281–305 (cit. on p. 55).
- [74] S. Bai, J. Z. Kolter, and V. Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. arXiv:1803.01271 [cs]. 2018. DOI: [10.48550/arXiv.1803.01271](https://doi.org/10.48550/arXiv.1803.01271). URL: <http://arxiv.org/abs/1803.01271> (cit. on p. 55).
- [75] R. Kohavi. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *Proceedings of the 14th international joint conference*

on Artificial intelligence - Volume 2. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143 (cit. on p. 57).

Acknowledgements

The author would like to sincerely thank Prof. Dr. Alessandro Del Vecchio for his invaluable mentorship and consistent encouragement throughout the course of this thesis. His deep understanding of both engineering principles and clinical applications significantly enriched the research process. The clarity and perspective he brought to every discussion greatly influenced the direction of this work and contributed meaningfully to the author's academic development. His high standards for scientific rigor, combined with his approachable guidance, created an environment that challenged the author to grow intellectually and professionally.

Special thanks are extended to Vlad Cnejevici for his dedicated guidance, technical support, and constructive feedback. His expertise in signal processing and machine learning was instrumental to the success of this project and helped the author navigate several critical stages of the work. His readiness to assist with both conceptual and implementation challenges, along with his clear and practical insights, greatly contributed to the overall quality and direction of this thesis.

The author would also like to thank Prof. Dr. Anne Koelewijn for granting permission to conduct the treadmill experiments, Maria Eleni Athanasiadou for her assistance during those sessions.

The author also gratefully acknowledges the anonymous participants who volunteered for the experimental sessions. Their time, effort, and cooperation were essential to the smooth execution of the study and are deeply appreciated.

Deepest thanks are reserved for Dhorali, Thamarai Selvi, Hariharan, and Akilandeswari for their unwavering love, patience, and encouragement throughout this academic journey. Their constant emotional support has been a foundation of strength and perseverance.

The author further extends his gratitude to all friends and relatives whose kind words, understanding, and motivation played an important role in making this work possible.

Figure 4.1 was designed using BioRender.com and is used with permission.

Declaration of authorship

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich odersinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Richtlinien des Lehrstuhls für Bachelor- und Masterarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Erlangen, 22.5.2025

Sathyanaarayanan Dhorali