Deep Learning for Enhanced Prosthetic Control: Real-Time Motor Intent Decoding for Simultaneous Control of Artificial Limbs

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Deep learning for enhanced prosthetic control: Real-time motor intent decoding for simultaneous control of artificial limbs

Jan Zbinden, Student Member, IEEE, Julia Molin, Max Ortiz-Catalan, Senior Member, IEEE

Abstract— The development of advanced prosthetic devices that can be seamlessly used during an individual's daily life remains a significant challenge in the field of rehabilitation engineering. This study compares the performance of deep learning architectures to shallow networks in decoding motor intent for prosthetic control using electromyography (EMG) signals. Four neural network architectures, including a feedforward neural network with one hidden layer, a feedforward neural network with multiple hidden layers, a temporal convolutional network, and a convolutional neural network with squeeze-and-excitation operations were evaluated in real-time, human-in-the-loop experiments with able-bodied participants and an individual with an amputation. Our results demonstrate that deep learning architectures outperform shallow networks in decoding motor intent, with representation learning effectively extracting underlying motor control information from EMG signals. Furthermore, the observed performance improvements by using deep neural networks were consistent across both able-bodied and amputee participants. By employing deep neural networks instead of a shallow network, more reliable and precise control of a prosthesis can be achieved, which has the potential to significantly enhance prosthetic functionality and improve the quality of life for individuals with amputations.

Index Terms— deep learning, myoelectric, pattern recognition, prostheses, prosthetics, simultaneous control.

I. INTRODUCTION

THE loss of a limb can significantly impact an individual's quality of life by making even basic daily activities challenging. Prosthetic limbs have the potential to restore some function after an amputation, thereby facilitating navigation through everyday tasks.

A prevalent method for controlling the joints of a prosthetic arm involves interpreting the electromyography (EMG) signals from the remnant muscles after amputation. The development of such interpretation algorithms to decode the motor intent from EMG signals has been an active area of research for several years: Standard machine learning algorithms such as Linear Discriminant Analysis (LDA), Support Vector Machines (SVM), and Multi-Layer Perceptron (a subset of feed forward neural networks or FFNN), have been successfully employed to control (virtual) myoelectric prostheses [1]–[7].

Deep learning, a more recent development in machine learning, utilizes neural networks with multiple layers to learn hierarchical representations of complex data and thereby solve pattern recognition and decision-making tasks. Whereas this approach often yields superior predictive accuracy compared to standard machine learning, it demands significantly more data, involves greater computational complexity, and necessitates careful hyperparameter tuning. Despite these challenges, deep learning-based neural networks have been remarkably successful in many applications, thus recent studies have begun exploring whether deep learning could enhance the prosthetic control performance of standard machine learning algorithms. For instance, Wand *et al.* [8], demonstrated that FFNNs, particularly those with multiple hidden layers, exhibited higher offline performance compared to the LDA approach.

One advantage of certain deep learning network architectures like Convolutional Neural Networks (CNNs) over standard machine learning algorithms is the elimination of feature-engineering reliance. Feature-engineering in the field of myoelectric control involves transforming raw EMG signals into lower-dimensionality features, reducing complexity and computational requirements. CNNs are inherently designed to extract meaningful features from raw data, a process known as representation learning. The absence of hand-crafted features could potentially allow networks to discover better data representations than those based on expert knowledge, as has occurred in fields such as image recognition [9], natural language processing [10], and speech recognition [11]. Several recent studies have shown that CNNs can effectively extract underlying motor control information from EMG signals in both offline data sets [12]–[14] and real-time control scenarios [15]. Combining CNNs feature-extracting architectures with other network modules (e.g., Recurrent Neural Networks (RNN) [16], Long Short-Term Memory networks (LSTM) [17], [18], or Temporal Convolutional Networks (TCN) [19], [20])

has resulted in even higher offline movement decoding performance.

Neural networks hold another advantage over standard machine learning algorithms for prosthetic control: their inherent capability for simultaneous classification [1]. This means that multiple joints can be active at once, rather than just one at a time. When using a prosthetic hand with individually actuated fingers to perform common grasp patterns [21], simultaneous classification becomes crucial, as closing fingers sequentially rather than simultaneously to grasp an object is slow, unintuitive, and frustrating.

In this study, we further explored deep neural networks for simultaneous prosthetic control. We evaluated the real-time, human-in-the-loop performance of four neural networks: a single hidden layer FFNN, a multi-hidden layer FFNN, a TCN, and a CNN with channel attention by Squeeze and Excitation (SE) operations (CNN-SE). A CNN-SE architecture was selected because it demonstrated significant improvement over several of the state-of-the-art CNNs [22] and promising results in bio-signal classification tasks (i.e., electrocardiogram [23], [24] and electroencephalogram [25], [26]), and recently also for offline EMG gesture recognition [27], [28]. The choice of the other three networks allowed us to investigate if the promising offline results of increasing the number of hidden layers in FFNNs [8] and the reportedly high performance of TCN's [19], [20] would translate to an online control scenario.

Our results indicated that deep learning architectures outperform shallow networks in decoding motor intent. All findings were obtained during human-in-the-loop experiments, i.e., the participants received visual feedback about the current prediction in real-time. This allowed them to adjust their muscle contractions to influence the next prediction. This step is crucial because offline algorithm performance - metrics based on algorithms trained and tested solely on prerecorded data - may not necessarily translate to effective real-time prosthesis control [3]. We further observed generalizable network performance in simultaneous control over three hand and wrist joints and simultaneous control over three fingers. Finally, we also observed a trend of improved motor intent decoding performance with the participant with amputation when using deep neural networks compared to a shallow network. These findings indicate a promising potential of utilizing such approaches for real-world prosthetic use.

II. METHODS

A. Study design

This study investigated if the use of deep neural network architectures can lead to a more accurate decoding of motor intent in the context of prosthetic control.

The main study objective was to compare different deep neural networks to assess their real-time performance (as measured by the *Motion Test*) in decoding individual and simultaneous motor intent for controlling a virtual prosthetic limb compared to a shallow network.

The secondary objective was to assess if any performance differences between networks are generalizable and persist between different sets of performed movements (as assessed by conduction of the *Motion Test* using either gross movements or finger movements).

The third objective was to investigate if any potential performance improvements between networks would translate from an able-bodied cohort to a participant with upper limb amputation.

The study protocols were carried out in accordance with the declaration of Helsinki. The signed informed consent was obtained before conducting the experiments. The study was approved by the Regional Ethical Review Board in Gothenburg (Dnr. T688-12, Dnr. 2022-06513-01, and Dnr. 2020-04600).

B. Experimental conditions

To investigate the three different pre-specified objectives, we compared the performance of four different neural network architectures, performed on two different movement sets and two cohorts.

The four neural network architectures consisted of a one-layer feed forwarded neural network, henceforth called FFNN1, a six-layer feed forward neural network (FFNN6), a temporal convolutional network (TCN), and a convolutional neural network with a squeeze and excitation module (CNN-SE).

For the first movement set (gross movements), we chose a commonly used set of movements [1] covering three degrees of freedom: hand open and close, wrist pronation and supination, and wrist flexion and extension. The gross movement condition consisted of all possible combinations of the three DoFs, totaling to 26 recorded movements (6, 12, and 8 active movements involving 1, 2, and 3 DoF, respectively). The second movement set (finger movements) consisted of thumb flexion and extension, index flexion and extension, and flexion and extension of a fused middle, ring, and little finger. Additional to the six 1-DoF movements, the finger movement condition consisted of a subset of all possible combinations to focus on common grasp patterns [21]: simultaneous flexion and extension of the thumb and index (pinch), simultaneous flexion and extension of the index and middle, ring, and little finger (prismatic grasp), and simultaneous flexion and extension of all three DoF (hand open and close). The total number of actively recorded movements was 12. In both conditions, EMG during a resting state was also recorded.

For the able-bodied cohort, ten participants were recruited on a convenience sample basis (2 male/8 female, mean age 26.5 and range 24-32) to perform the *Motion Test* using gross movements. Ten able-bodied participants were recruited on a convenience sample basis (6 male/4 female, mean age 28.5 and range 25-33) to perform the *Motion Test* using finger movements. The number of participants was chosen based on a power analysis assuming a 10% score increase at 0.1 standard deviation.

One participant with transhumeral amputation, age 54, who received an implanted neuromusculoskeletal interface with electro-neuromuscular constructs [29] was recruited for both the gross and finger movement condition. This specific participant was recruited as the surgically created electro-neuromuscular constructs allowed him to perform both the gross and finger movements.

C. Setup for able-bodied participants

Eight pair of surface electrodes in a bipolar configuration were placed around the arm with an interelectrode distance of 2 cm. The first pair were targeted towards *extensor carpi ulnaris.* For the gross movement condition, the other seven electrode pairs were equally spaced around the arm. For the finger movement condition, five other pairs were equally spaced around the forearm and two pairs were targeted towards *Flexor pollicis longus* and *Extensor Indicis*, respectively. Additionally, one electrode was placed on the *Ulnar Styloid* as reference.

The EMG-signals were acquired with an in-house developed system [30]. Standard parameters [31] for acquiring (sampling rate of 1,000 Hz) and filtering (an analogue low-pass filter with cutoff frequency 500 Hz, and a digital butterworth high-pass filter with cutoff frequency of 20 Hz and a second order notch filter at 50 Hz) surface EMG signals were used.

D. Setup for a participant with transhumeral amputation

One participant with a transhumeral amputation who received an implanted neuromusculoskeletal interface [32] was recruited for the evaluation in potential users. The participant underwent a nerve transfer surgery to create myoelectric sources for joints lost due to the amputation [29]. Specifically, three native muscles and five free muscles grafts were reinnervated by the ulnar, median, and radial nerve, allowing for control of individual fingers. Instead of using Ag/AgCl surface electrodes, the electrodes were implanted on the epimysium or intramuscularly to simplify daily use of the prosthesis (the electrodes stay at the same place, so no daily retraining of the control algorithm is necessary), allow signal acquisition from all electro-neuromuscular constructs (the free muscle grafts are placed deep inside the arm, thus their signals are difficult to acquire with surface electrodes), and generally improve the acquired signals (higher signal-to-noise ration due to measuring closer to the signals source). Together with four electrodes on unreconstructed muscles, signals of a total of 12 monopolar implanted electrodes were used. The abutment of the osseointegrated implant was used as reference.

The signals from the electrodes were acquired using the Artificial Limb Controller, an embedded system for controlling prosthetic devices [33]. The signals were sampled at 500 Hz (limited by the available embedded system memory) with 16-bit resolution and online high-pass and notch filtered at 20 Hz and 50 Hz, respectively.

E. Training dataset acquisition and preprocessing

The participants were instructed to perform each movement at contraction levels of around 50-70% of their maximum voluntary contraction strength. Each movement was recorded for 5s each, repeated 6 times. The first and last 10% of each recording was removed to exclude the transient period of the contraction. Each recording was further divided into windows with a time length of 200 ms, using an overlap of 150 ms. The second and fifth recording of every movement were reserved to be used as test- and validation data. The remaining recordings represented the training dataset. Due to the data being recorded in windows with overlapping information, using specific recordings instead of a randomized split of all windows of data avoids having trained on data in the test and validation sets.

Four (mean absolute value, zero crossing, slope sign changes, and wave form length) of the five features in the Hudgins set [34] were extracted channel-wise for the feed forward networks, whereas no features were extracted for the convolutional architectures. Finally, the datasets were normalized by z-score normalization. The normalization occurred channel-wise for the convolutional architectures, whereas feature-wise for each channel for the feed forward architectures. To keep the test- and validation datasets unseen, these were normalized using the mean- and standard deviation values from the training set solely.

F. Neural networks

The FFNN1 consisted of one FFNN module (see Fig. 1a), which contains a fully connected layer with 128 nodes, a dropout layer (drop out probability $d = 0.1$), and rectified linear unit (ReLU) layer.

The FFNN6 consisted of six FFNN modules, with the same 128 nodes in each fully connected layer and drop out probability of 10%.

The TCN architecture consisted of three consecutive TCN modules (see Fig. 1b). The module included two 1D convolutions with 64 filters with a size of 8. Each convolutional layer was followed by layer normalization. A spatial dropout ($d = 0.005$) layer was applied after the first normalization layer. The module ended with a ReLU activation layer, followed by an addition layer to provide residual connection. The last TCN module applied global average pooling before its output module.

The CNN-SE architecture consisted of four Conv-SE Modules (see Fig. 1c), with 1D convolutional filters with sizes of 20, 5, 3, and 3 respectively. The first CNN-SE module did not apply max pooling, whereas the other 3 modules had 1D max pooling using sizes of 5, 3, and 2, respectively. The max pooling was applied with a stride equal to its operational size. To avoid overfitting, a spatial dropout layer ($d = 0.1$) was used. Every Conv-SE module ended with a Squeeze and Excitation (SE) block, aiming to give attention to the feature maps. All convolutional layers used 64 number of filters. The last Conv-SE module was followed by two FFNN modules (Fig. 1a) with 128 nodes in their fully connected layers. All networks ended with an Output Module (Fig. 1d), with 7 neurons in its fully connected layer, corresponding to both directions of each of the 3DoF movements and the "Rest" class. Since all four models had sigmoidal outputs in range [0,1] to allow for multi-label classification, a threshold of 0.85 was set consistently on all classes for all models.

The number of feed forward layers in the FFNN6, CNN-SE and TCN were chosen based on our pre-study (see Appendix 5 and 6). All networks were implemented and trained using Matlab 2021b.

G. Model Fitting

All models were trained offline using the Adam [34] optimizer with Binary Cross Entropy (BCE) as loss function.
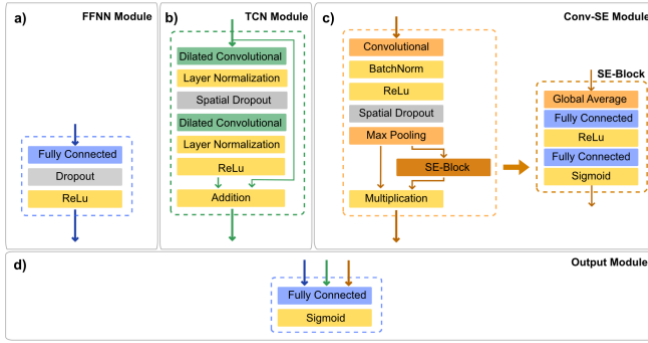
Fig. 1. Overview of the used neural network architecture modules. a) The feed forward neural network (FFNN) module, used in the FFNN1 and FFNN6 networks, and at the end of the CNN-SE network. b) The Temporal Convolutional Networks (TCN) module, main building block of the TCN architecture. c) Convolutional Squeeze and Excitation (Conv-SE) module, main building block of CNN-SE network, with zoom into the Squeeze and Excitation (SE) block that improves network generalizability by recalibrating channel-wise feature responses by taking dependencies between channels into account. d) Output Module with 7 (number of classes) nodes in its fully connected layer and a sigmoid layer that allows for multi-label classification. Used as final layers in all four networks.

The CNN-SE and TCN both had a mini-batch size of 128, iterating for at most 100 epochs. Both the FFNN1 and FFNN6 used all training samples in every batch and iterated for at most 1,000 epochs. The training was terminated if the validation loss didn't improve for 5 consecutive iterations to prevent overfitting, known as early-stopping. The initial learning rate was 0.001 and decayed by a factor of 0.1 every third epoch for the CNN and TCN, whereas for the FFNN1 and FFNN6 it remained constant. All models further adopted L2 regularization with its lambda hyperparameter set to 0.01.

H. Motion Test

The real time performance of the algorithms was evaluated through the *Motion Test* protocol, first introduced by Kuiken *et al.* [35]. In this test, the participants seated comfortably in front of a computer (see Fig. 2a) and asked to perform movements randomly prompted on a screen (see Fig. 2b).

In both the gross and finger movement condition, the list of active movement was randomized using a uniform pseudorandom number generator prior to each *Motion Test* trial. A total of three consecutive *Motion Test* trials were performed for each of the four neural networks and per condition. A movement was considered complete if a total of 2s (40 predictions) of the correct output was predicted within the time limit of 10s. Once a movement completed or reached its time limit, a new movement was requested. The four neural networks were tested in a randomized order and the participants were not told which of the four neural networks was in use.

Four real-time performance metrics were calculated to assess the *Motion Tests* outcomes. The Completion Rate (as the percentage of completed motions), the Completion Time (as the time between the first correct prediction and the completion of the motion), the Exact Match Ratio (EMR) (the ratio of exact correct output vectors over all predicted output vectors), and the F1-score (harmonic mean of precision and recall).
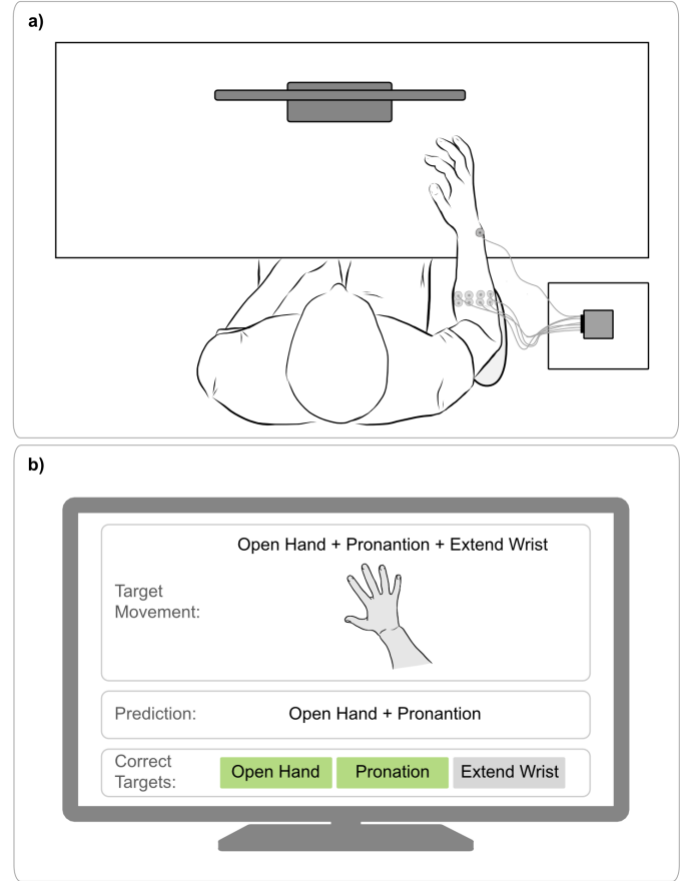

Fig. 2. Motion Test setup. (A) Able-bodied participant sitting in front of monitor while being connected to an EMG acquisition system. The gross movement condition electrode configuration is depicted, with eight bipolar surface electrode pairs placed around the arm and one electrode placed on the Ulnar Styloid as reference. (B) The Motion Test interface, divided into three sections: the top section describes the demanded movement in text and image which remained static for a certain movement, the middle section shows the predicted movement which were updated every 50ms, and the bottom section shows the targets which colors changed to green if the demanded target was predicted.

The EMR is calculated as follows:

$$EMR = \frac{1}{N}\sum_{t=1}^{N}\left(\hat{y}_{tk} = y_{tk}, \forall k\right); \quad EMR \in [0,1]$$

where $\hat{y}$ denotes the predicted outputs, y the true outputs, and $\hat{y}_{tk}$ and y_{tk} the t'th output of class k. *N denotes the total number of outputs.*

The F-1 score is defined as:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}; \quad F1 \in [0,1]$$

Where $precision$ is the ratio of true positives over all positive predicted outputs of class k and $recall$ is the ratio of true positives among all instances that belong to class k.

Both the Exact Match Ratio and the F1- score are metrics often used for multi-label classification [36] and replace the conventionally used accuracy in multi-class classification tasks. One movement might have lower completion time than another during the motion test, and thereby have fewer predictions to be compared with - which result in class imbalances. To remove as much bias as possible from this imbalance, the EMR is calculated as the macro average over the EMR of each demanded movement.

A common way to extract metrics of the motion test is to only consider the predictions during the completion time. Thus, only completed movement would contribute to the evaluation metrics derived during this time. This practice skews the results as the misclassifications of the worst performing classes, i.e., the ones where the movements is not completed and by extension deteriorate prosthetic functionality the most, are neglected. To get a more complete understanding of the classifier performance, we derived the EMR and F1-Score from all predications made during the test, regardless of completion.

I. Data analysis/ Statistics

The Wilcxon Signed rank test was used to determine statistical significance ($p<0.05$) between the *Motion Test* outcomes of the shallow network (FFNN1) and the deep neural networks (FFNN6, TCN, CNN-SE) for able-bodied participants.

III. RESULTS

The result is presented in boxplots where the bottom and top edges indicate the 25th and 75th percentiles, respectively. Outliers are represented as "+", and statistical significance

($p<0.05$) are shown by "*". Median values are shown as a horizontal line across each corresponding box.

A. Motion test outcomes from able-bodied participants

Decoding the individual and simultaneous movements of seven gross movements (hand open/close, wrist pro/supination, wrist flexion/extension, and no movement (rest)) showed that deep neural network architectures perform better during the *Motion Test* than shallow networks. The deep neural networks (FFNN6, TCN, and CNN-SE) achieved around 5% higher completion rates compared to the shallow network (FFNN1), with the CNN-SE featuring the highest completion rate, completing 71 out of the 78 demanded movements (see Fig. 3a and Appendix 1). The completion times decreased by 9-11% when using the deep neural network compared to the FFNN1 (see Fig. 3b). The EMR improved by 8-12%, while the F1-score improved by 2-3% using the deep neural networks compared to the shallow network (see Fig. 3c and Fig. 3d). For individual movements, the shallow network performs up to 10% better in terms of completion rate and EMR score compared to the deep neural networks (see Fig. 3e-h). However, for simultaneous movements, the deep neural networks reach up to 30% higher completion rates, 22% lower completion times, 50% higher EMR scores, and 9% higher F1-scores (see Fig. 3e-h). Similar trends as in the gross movement condition were observed for conducting the *Motion Test* with individual and simultaneous finger movements (thumb flexion/extension, index flexion/extension, and middle/ring/little finger flexion/extension). The deep neural networks achieved up to 17% higher completion rates and 19% lower completion times (see Fig. 4a, Fig. 4b, and Appendix 2). And the EMR improved between 17-36% and the F1-Score between 9-16% when using

Motion test outcomes gross movements
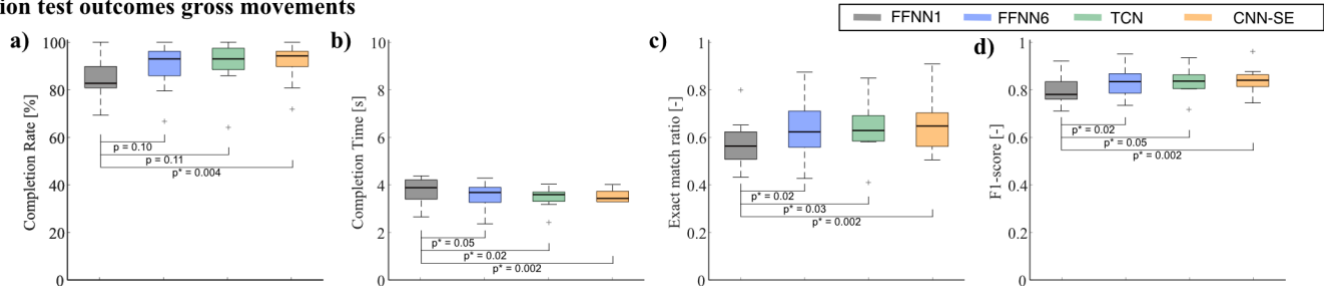
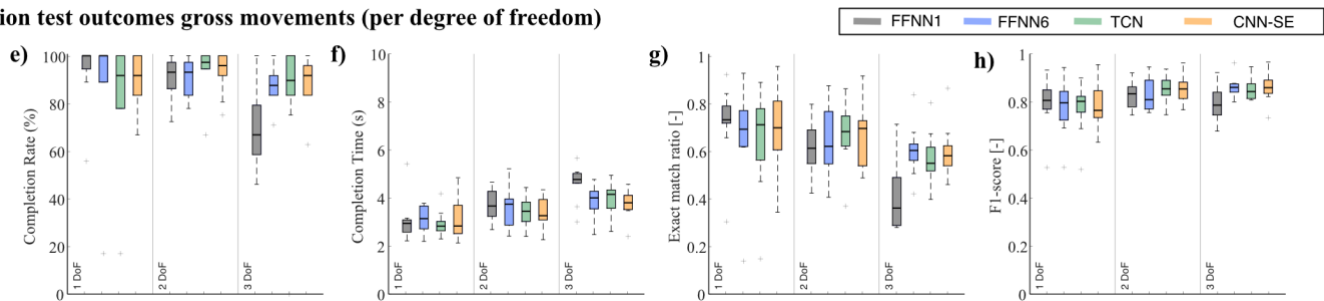**Motion test outcomes gross movements (per degree of freedom)**

Fig. 3. Motion test outcomes of able-bodied participants (n = 10 participants) controlling simultaneous hand open/close, wrist pronation/supination, and wrist flexion/extension movements (n = 26 movements, 3 trials). (a) Motion test completion rate for simultaneous gross movements where 100% indicates that all required movements were performed. (b) Motion test completion time (c) Exact-match-ratio and (d) F1-score for all degrees of freedom during the motion test. (e-f) Show all metrics split up per degree of freedom (1DoF n = 6 movements, 2DoF n = 12 movements, and 3DoF n= 8 movements).

Motion test outcomes single finger movements
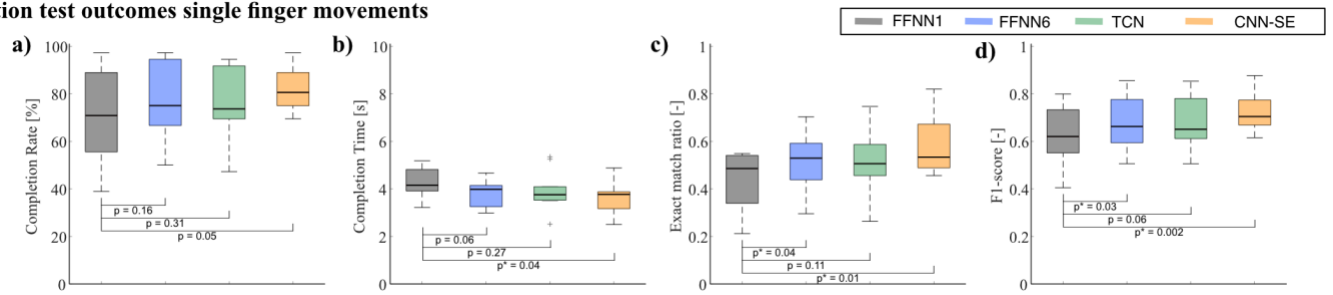
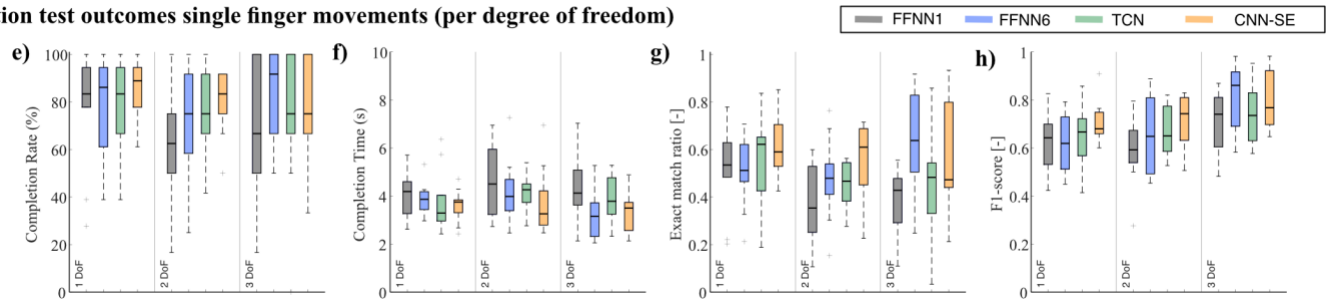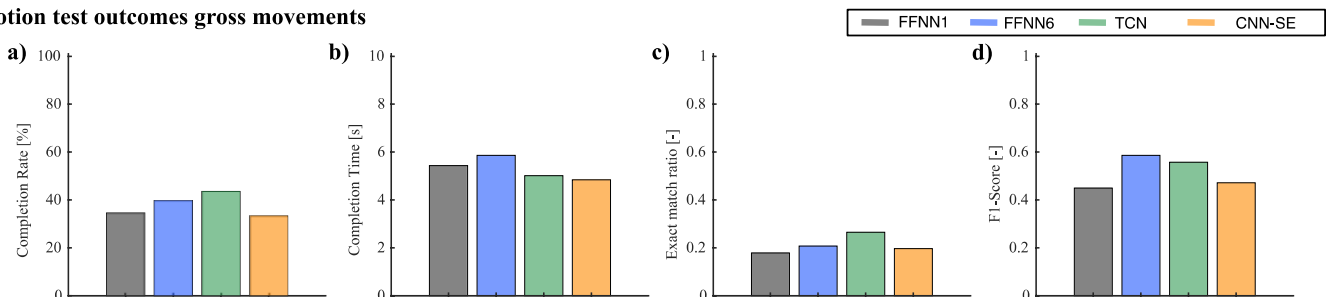**Motion test outcomes single finger movements (per degree of freedom)**

Fig. 4. Motion test outcomes of able-bodied participants (n = 10 participants) controlling simultaneous thumb flexion/extension, index flexion/extension, and middle/ring/little finger flexion/extension movements (n = 12 movements, 3 trials). (a) Motion test completion rate for simultaneous finger movements. (b) Motion test completion time (c) Exact-match-ratio and (d) F1-score for all degrees of freedom during the motion test. (e-f) Show all metrics split up per degree of freedom (1DoF n = 6 movements, 2DoF n = 4 movements, and 3DoF n= 2 movements).

deep neural network architectures compared to a shallow network (see Fig. 4c and Fig. 4d). In the finger movement case, however, the shallow network was generally outperformed by deep neural networks during both individual (e.g., up to 17% higher EMR score and 13% higher F1-score) and simultaneous (e.g., up to 65% higher EMR score and 20% higher F1-score) movements (see Fig. 4e-h).

B. Motion test outcomes from participant with amputation

For the participant with amputation, completion rates of 34.6% (FFNN1), 39.7% (FFNN6), 43.6% (TCN), and 33.3% (CNN-SE) were observed during the *Motion Tests* with gross movements (see Fig. 5a and Appendix 3). The corresponding completion times were: 5.4s (FFNN1), 5.8s (FFNN6), 5.0s (TCN), and 4.8s (CNN-SE) (see Fig. 5b).

Motion test outcomes gross movements

Motion test outcomes gross movements (per degree of freedom
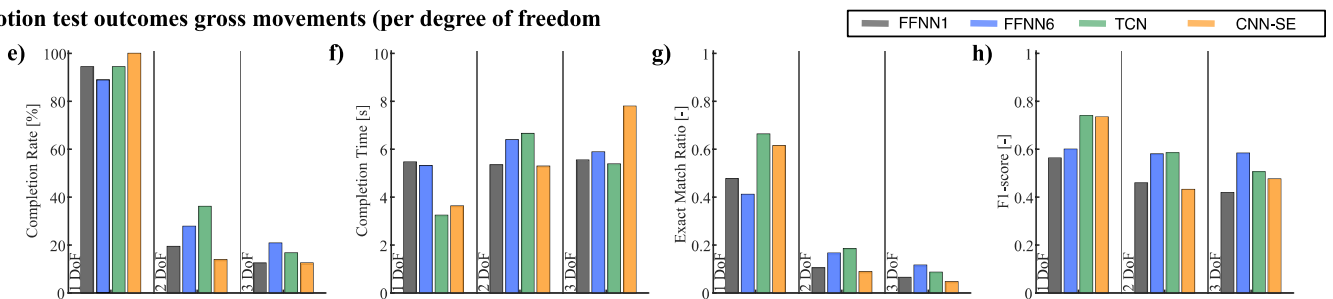
Fig. 5. Motion test outcomes of a participant with amputation (n = 1) controlling simultaneous hand open/close, wrist pronation/supination, and wrist flexion/extension movements (n = 26 movements, 3 trials). (a) Motion test completion rate for simultaneous gross movements. (b) Completion time of Motion Test (d) Exact-match- ratio and (d) F1-score for all degrees of freedom during the motion test. (e-f) Show all metrics split up per degree of freedom (1DoF n = 6 movements, 2DoF n = 12 movements, and 3DoF n= 8 movements). The network architectures were tested in the following order: FFNN6, TCN, FFNN1, CNN-SE.

Motion test outcomes finger movements

Motion test outcomes finger movements (per degree of freedom

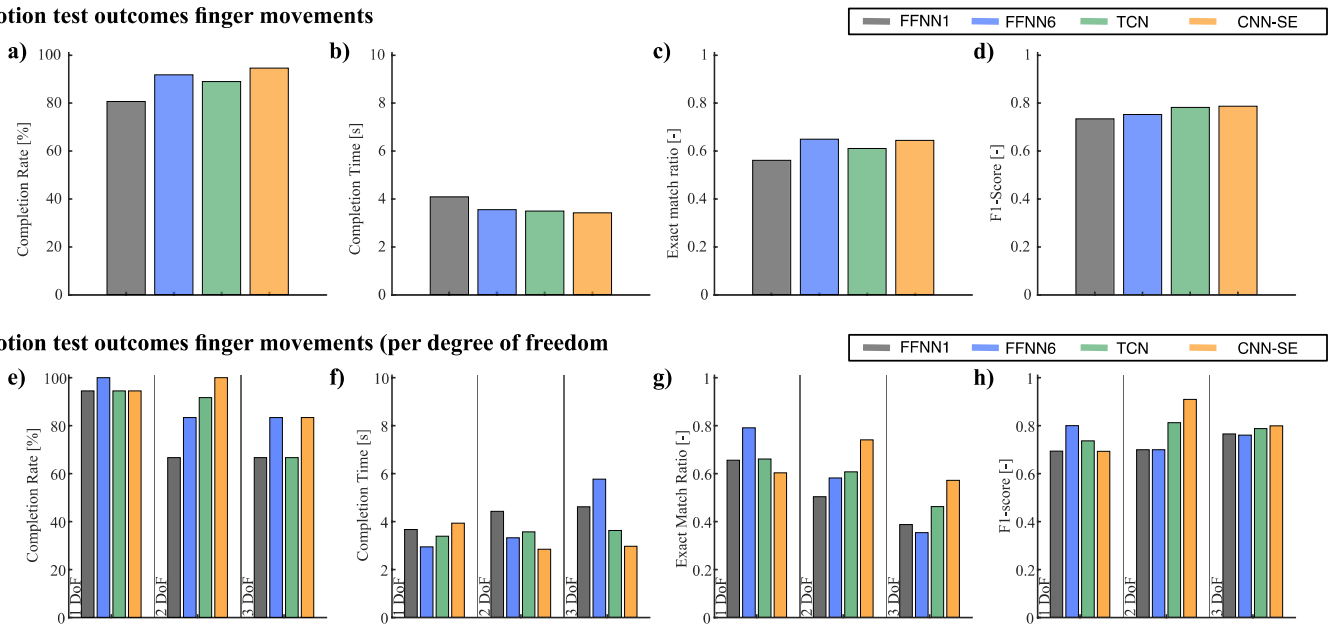

Fig. 6. Motion test outcomes of a participant with amputation (n = 1) controlling thumb flexion/extension, index flexion/extension, and middle/ring/little finger flexion/extension movements (n = 12 movements, 3 trials). (a) Motion test completion rate for simultaneous gross movements. (b) Completion time of Motion Test. (c) Exact-match-ratio and (d) F1-score for all degrees of freedom during the motion test. (e-f) Show all metrics split up per degree of freedom (1DoF n = 6 movements, 2DoF n = 4 movements, and 3DoF n= 2 movements). The network architectures were tested in the following order: TCN, FFNN6, FFNN1, CNN-SE.

The respective EMR and F1-scores were 0.18 and 0.45 (FFNN1), 0.21 and 0.59 (FFNN6), 0.27 and 0.56 (TCN), and 0.20 and 0.47 (CNN-SE) (see Fig. 5c and Fig. 5d). A considerable difference in completion rate and EMR score was observed between individual and simultaneous movements for all four networks (see Fig. 5e-h).

During the Motion Test with finger movements, performed by the participant with amputation, completion rates increased between 10-17% from 80.6% (FFNN1) to 91.6% (FFNN6), 88.9% (TCN), and 94.4% (CNN-SE) (see Fig. 6a and Appendix 4). The corresponding completion times decreased by around 15% from 4.0s (FFNN1) to 3.5s (FFNN6), 3.5s (TCN), and 3.4s (CNN-SE) (see Fig. 6b). The EMR increased by 9-14% from 0.56 (FFNN1) to 0.64 (FFNN6), 0.61 (TCN), and 0.64 (CNN-SE) (see Fig. 6c). The F1 scores increased by 3-8% from 0.73 (FFNN1) to 0.75 (FFNN6), 0.78 (TCN), and 0.79 (CNN-SE) (see Fig. 6d). Like the previous conditions, the deep neural networks generally performed better, especially during simultaneous movements. E.g., 1DoF completion rate was similar for all networks, but 2DoF and 3DoF completion rate increased by 25% and 50%, respectively (see Fig. 6e-h)

IV. DISCUSSION

In this study, we investigated four neural network architectures in real-time to determine their capability of decoding motor intention, with the aim of finding an algorithm that could ultimately lead to more functional control of a prosthetic limb.

The *Motion Test* results with able-bodied participants controlling simultaneous gross hand movements showed that deep neural networks outperformed shallow networks, particularly when performing simultaneous movements. The deep neural networks generally surpassed the FFNN1 in completion rate, completion time, EMR, and F1-Score. This could be attributed to the fact that deep networks have a higher capacity to learn complex representations of the input data, enabling them to effectively capture the dependencies between multiple movements. The one-layer network struggled to learn these dependencies, resulting in more reliable predictions for sequential, and only partially correct predictions for simultaneous movements. This is most apparent when comparing the changes in EMR and F1-Scores: the deep neural networks led to a 10% increase in EMR (only counting movements as correct when they were predicted exactly as prompted) compared to the shallow network, while the F1 score (taking partially correct simultaneous movements into account) increased by just a few percent. A possible explanation of the specific preference of the shallow network to predict individual over simultaneous movements could be due to its lack of depth (prohibiting it to learn hierarchical representations) and the lower number of learnable parameters (possibly leading to overfitting on the simultaneous movements). Increasing the number of neurons, and thereby the number of learnable parameters, was found to increase myoelectric decoding performance of shallow networks, however with diminishing returns for networks with over 32 neurons and no performance gains for networks with over 128 neurons [8]. In the same study, it was shown that networks with additional hidden layers outperform single-layer networks. This suggests that in our case, hierarchical learning had a stronger contribution on decoding performance for more complex tasks than the number of learnable parameters. Nevertheless, it is still possible that a shallow network with a more complex architectures than simply

a fully connected layer could display high motor intent decoding performance, even for complex tasks, and should therefore be considered for future work.

When able-bodied participants were tasked with controlling simultaneous finger movements, the deep neural networks also achieved higher Motion Test scores compared to the shallow network. Predicting finger movements from muscles using surface electrodes requires decoding signals with lower amplitude and more crosstalk than those from larger muscle groups – making it a more complex task than decoding gross movements. The greater difference in scores between the shallow and deep network architectures in the finger compared to the gross movement condition suggests that deep neural networks are remarkably more beneficial for more complex tasks.

Similar trends were observed when conducting the same experiments with a participant with amputation. However, this participant achieved higher scores in finger movements compared to gross movements. This outcome, despite being the opposite of the outcome of the able-bodied group, was not entirely unexpected. Already during the recording of the ground truth data, the participant reported difficulties in reliably producing several of the simultaneous gross movements with their phantom hand. Indeed, decoding simultaneous movements during the Motion Test was not reliable and lead to the participant being frustrated. Consequently, simultaneous movement scores were substantially lower compared to the individual movement scores. The lower performance of the CNN-SE, which in all other cases performed best, could be explained by the fact that it was tested as the last of the four conditions and the participant lost focus due to long experiments demanding movements they could not produce. Contrastingly, the participant demonstrated better control of their phantom fingers, producing reliable and repeatable signals for both individual and simultaneous finger movements.

The experiments conducted in this study confirm previous results [12], [15], [37] that representation learning (as used in the TCN and CNN-SE) can extract features at least as well as engineered features (used in the FFNN6). Contrary to our expectations, representation learning did not lead to a distinctive performance improvement compared to the four engineered Hudgins features [34] used in this study and employed on our embedded system [33] used by patients who received a neuromusculoskeletal interface in daily life [29], [38], [39]. Thus, future experiments with larger ground truth dataset (to avoid potential overfitting and improve feature learning) and more complex tasks (decoding more intricate movements could potentially benefit from networks with a higher number of learnable parameters) will be required to determine if the added complexity of a network architecture capable of representation learning is justified. Particularly when deploying algorithms in self-contained prosthetic systems [29], [38] that must run on resource-constrained system [33], the considerably higher memory footprint and computational cost of a complex deep neural network might not be a worthwhile trad-off for a slightly better performance.

Surface EMG is known to be susceptible to motion artifacts and electromagnetic interference, which are problems less frequently found with implanted electrodes, thus making the latter more reliable for prosthetic control in daily life [38]. Implanted electrodes can provide access to myoelectric sites that are not accessible from the surface of the skin, such as deeper muscles or reinnervated free muscle grafts [29], [39] and thus facilitate the motor decoding task. In this regard, it could be argued that more electrodes are generally preferred, and thus the use of high-density arrays [40] could improve our results. However, one must consider where such arrays are placed as natively innervated muscles will not necessarily deliver more information, and mixed- or hyper-reinnervated muscles probably require implanted solutions to overcome the exponential complications, e.g. electrode lift-off, of having several surface electrodes.

The main limitation of this study is that only one participant with amputation participated in our experiments. Additionally, we did not assess how much human learning affected the motion intent decoding performance. During human-in-the-loop experiments, and also during daily prosthetic use, people learn to adapt their muscle contractions and thereby their myoelectric signals to the network they use - which can greatly affect the control performance. Randomizing the network order reduced the confounding effect of learning in our study. However, the learning effect is potentially still strong enough for us to refrain on speculating on the contribution of the different deep network architectures, given their similar performances. More rigorous analysis of network architectures to, for example, evaluate if certain architectures facilitate human learning and thereby would lead to improved performance over time is thus needed. Approaches like incremental learning [41] or reinforcement learning [42], [43] which incorporating the human actively during training the network could further improve performance: the network training would reflect the real-use environment more closely and incorporates human learning to fine-tune the network to individual preferences. An additional limitation is, that this study focused on decoding motion intent, without assessing controllability (how well a prosthesis can be controlled) or functionality (how much a prosthesis enhances function for daily life activities). Both are important metrics to consider when translating prosthetic technology from the laboratory to home-use. To improve controllability and possibly functionality, the networks can be fine-tuned (e.g., individual thresholds for each class or manual adjustment of the bias in the final network layer) and be supplemented by post-processing algorithms [44], [45].

The broader implications of our findings have significant potential to improve the field of prosthetic control and impact the lives of individuals with limb loss. By identifying effective neural network architectures for decoding motor intention, we can develop more advanced and user-friendly prosthetic devices that can restore a higher level of functionality for users. This, in turn, could lead to better integration of prosthetic limbs into daily life activities, enhancing the independence and overall quality of life for individuals with amputations. Furthermore, the insights gained from our study regarding the benefits of deep neural networks for decoding complex motor tasks might also have applications in other areas, such as rehabilitation robotics or brain-computer interfaces, where

accurate decoding of motor intent is crucial for effective human-machine interaction.

V. CONCLUSION

In this study, we established that deep learning architectures generally provide superior performance in decoding motor intent compared to shallow networks for both able-bodied participants and an individual with an amputation. We observed the biggest performances differences between deep and shallow networks during the complex task of decoding simultaneous movements. However, there are certain conditions, i.e. individual 1 DoF movements, where the shallow network performs equivalently if not better than deep networks. Our findings indicate that representation learning can effectively extract underlying motor control information from EMG signals to decode motor intent in real-time. By adopting deep neural networks as an alternative to shallow networks, a more reliable and precise control of a prosthesis could be achieved. Ultimately, this enhanced control has the potential to help restore some of the lost functionality resulting from amputation, and thus significantly improving the quality of life for affected individuals.

APPENDIX

A1 Table Overview of Motion Test result of able-bodied participants controlling 3 degrees of freedom gross movements.

A2 Table Overview of Motion Test result of able-bodied participants controlling 3 degrees of freedom finger movements.

A3 Table Overview of Motion Test result of the participant with amputation controlling 3 degrees of freedom gross movements.

A4 Table Overview of Motion Test result of the participant with amputation controlling 3 degrees of freedom finger movements. The best performances per metric are highlighted as bold text.

A5 Offline study to determine number of feed forward layers.

A6 Table Overview of offline F1-scores depending on the number of network layers.

ACKNOWLEDGMENT

REFERENCES

[1] M. Ortiz-Catalan, B. Hakansson, and R. Branemark, "Real-Time and Simultaneous Control of Artificial Limbs Based on Pattern Recognition Algorithms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 756–764, Jul. 2014.

[2] A. Dellacasa Bellingegni *et al.*, "NLR, MLP, SVM, and LDA: a comparative analysis on EMG data from people with trans-radial amputation," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, p. 82, Dec. 2017.

[3] S. Abbaspour, A. Naber, M. Ortiz-catalan, H. Gholamhosseini, and M. Lindén, "Real-time and offline evaluation of myoelectric pattern recognition for the decoding of hand movements," *Sensors*, vol. 21, no. 16, 2021.

[4] A. J. Young, L. H. Smith, E. J. Rouse, and L. J. Hargrove, "A new hierarchical approach for simultaneous control of multi-joint powered prostheses," *Proc. IEEE RAS EMBS Int. Conf. Biomed. Robot. Biomechatronics*, pp. 514–520, 2012.

[5] A. Krasoulis and K. Nazarpour, "Myoelectric digit action decoding with multi-output, multi-class classification: an offline analysis," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, 2020.

[6] A. Ameri, E. N. Kamavuako, E. J. Scheme, K. B. Englehart, and P. A. Parker, "Support vector regression for improved real-time, simultaneous myoelectric control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 6, pp. 1198–1209, 2014.

[7] N. Jiang, K. Englehart, and P. Parker, "Extracting Simultaneous and Proportional Neural Control Information of Multiple Degree of Freedom from the Surface Electromyographic Signal," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1070–1080, 2009.

[8] M. Wand, M. Kirstoffersen, A. W. Franzke, and J. Schmidhuber, "Analysis of Neural Network based Proportional Myoelectric Hand Prosthesis Control," *IEEE Trans. Biomed. Eng.*, vol. 9294, no. c, pp. 1–1, 2022.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch," *Proc. - 2017 IEEE 3rd Int. Conf. Collab. Internet Comput. CIC 2017*, vol. 2017-Janua, pp. 328–338, Mar. 2011.

[11] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[12] A. E. Olsson, P. Sager, E. Andersson, A. Björkman, N. Malešević, and C. Antfolk, "Extraction of Multi-Labelled Movement Information from the Raw HD-sEMG Image with Time-Domain Depth," *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019.

[13] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Front. Neurosci.*, vol. 11, no. JUL, pp. 1–11, 2017.

[14] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,"

Front. Neurorobot., vol. 10, no. SEP, pp. 1–10, 2016.

[15] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "Real-time, simultaneous myoelectric control using a convolutional neural network," *PLoS One*, vol. 13, no. 9, p. e0203835, Sep. 2018.

[16] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, and W. Geng, "A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition," *PLoS One*, vol. 13, no. 10, p. e0206049, Oct. 2018.

[17] N. K. Karnam, S. R. Dubey, A. C. Turlapaty, and B. Gokaraju, "EMGHandNet: A hybrid CNN and Bi-LSTM architecture for hand activity classification using surface EMG signals," *Biocybern. Biomed. Eng.*, vol. 42, no. 1, pp. 325–340, 2022.

[18] X. Chen, Y. Li, R. Hu, X. Zhang, and X. Chen, "Hand Gesture Recognition based on Surface Electromyography using Convolutional Neural Network with Transfer Learning Method," *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 4, pp. 1292–1304, Apr. 2021.

[19] J. L. Betthauser *et al.*, "Stable Responsive EMG Sequence Prediction and Adaptive Reinforcement With Temporal Convolutional Networks," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 6, pp. 1707–1717, Jun. 2020.

[20] P. Tsinganos, B. Jansen, J. Cornelis, and A. Skodras, "Real-Time Analysis of Hand Gesture Recognition with Temporal Convolutional Networks," *Sensors*, vol. 22, no. 5, p. 1694, Feb. 2022.

[21] A. M. Dollar, "Classifying Human Hand Use and the Activities of Daily Living," in *Springer Tracts in Advanced Robotics*, vol. 95, R. Balasubramanian and V. J. Santos, Eds. Cham: Springer International Publishing, 2014, pp. 201–216.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 7132–7141, 2018.

[23] J. Park, J. K. Kim, S. Jung, Y. Gil, J. Il Choi, and H. S. Son, "ECG-signal multi-classification model based on squeeze-and-excitation residual neural networks," *Appl. Sci.*, vol. 10, no. 18, 2020.

[24] J. Cai, W. Sun, J. Guan, and I. You, "Multi-ECGNet for ECG Arrythmia Multi-Label Classification," *IEEE Access*, vol. 8, pp. 110848–110858, 2020.

[25] G. A. Altuwaijri, G. Muhammad, H. Altaheri, and M. Alsulaiman, "A Multi-Branch Convolutional Neural Network with Squeeze-and-Excitation Attention Blocks for EEG-Based Motor Imagery Signals Classification," *Diagnostics*, vol. 12, no. 4, p. 995, Apr. 2022.

[26] N. Ke, T. Lin, and Z. Lin, "Channel-Weighted Squeeze-and-Excitation Networks For Epileptic Seizure Detection," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021, vol. 2021-Novem, pp. 666–673.

[27] Z. Yang *et al.*, "Dynamic Gesture Recognition Using Surface EMG Signals Based on Multi-Stream Residual Network," *Front. Bioeng. Biotechnol.*, vol. 9, no. October, pp. 1–13, 2021.

[28] Y. Ma, C. Wang, C. Jiang, and Z. Li, "Deep Residual Shrinkage Networks for EMG-based Gesture Identification," pp. 1–5, Feb. 2022.

[29] J. Zbinden *et al.*, "Improved control of a prosthetic limb by surgically creating electro-neuromuscular constructs with implanted electrodes," *Sci. Transl. Med.*, vol. 15, no. 704, Jul. 2023.

[30] M. Buist, S. Damercheli, J. Zbinden, M. T. N. Truong, E. Mastinu, and M. Ortiz-Catalan, "Novel Wearable Device for Mindful Sensory Motor Training: Integrating Motor Execution and Somatosensory Stimulation for Sensorimotor Rehabilitation," *Under Rev.*, 2023.

[31] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface EMG signal: Movement artifact and baseline noise contamination," *J. Biomech.*, vol. 43, no. 8, pp. 1573–1579, 2010.

[32] M. Ortiz-Catalan, B. Hakansson, and R. Branemark, "An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs," *Sci. Transl. Med.*, vol. 6, no. 257, pp. 257re6-257re6, Oct. 2014.

[33] E. Mastinu, P. Doguet, Y. Botquin, B. Hakansson, and M. Ortiz-Catalan, "Embedded System for Prosthetic Control Using Implanted Neuromuscular Interfaces Accessed Via an Osseointegrated Implant," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 4, pp. 867–877, Aug. 2017.

[34] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, 1993.

[35] T. A. Kuiken, "Targeted Muscle Reinnervation for Real-time Myoelectric Control of Multifunction Artificial Arms," *JAMA*, vol. 301, no. 6, p. 619, Feb. 2009.

[36] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[37] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "Regression convolutional neural network for improved simultaneous EMG control," *J. Neural Eng.*, vol. 16, no. 3, 2019.

[38] M. Ortiz-Catalan, E. Mastinu, P. Sassu, O. Aszmann, and R. Brånemark, "Self-Contained Neuromusculoskeletal Arm Prostheses," *N. Engl. J. Med.*, vol. 382, no. 18, pp. 1732–1738, Apr. 2020.

[39] M. Ortiz-Catalan *et al.*, "A highly integrated bionic hand with neural control and feedback for use in daily life," *Sci. Robot.*, vol. 8, no. 83, pp. 1–9, 2023.

[40] M. J. Zwarts and D. F. Stegeman, "Multichannel surface EMG: Basic aspects and clinical utility," *Muscle Nerve*, vol. 28, no. 1, pp. 1–17, Jul. 2003.

[41] A. Gijsberts *et al.*, "Stable myoelectric control of a hand prosthesis using non-linear incremental learning," *Front. Neurorobot.*, vol. 8, no. FEB, pp. 1–15, 2014.

[42] K. Freitag, R. Laezza, J. Zbinden, and M. Ortiz-Catalan, "Improving Bionic Limb Control through Reinforcement Learning in an Interactive Game

Environment," *Interact. Learn. with Implicit Hum. Feed. Work. 40 th Int. Conf. Mach. Learn. Honolulu, Hawaii, USA. PMLR 202, 2023.*, 2023.

[43]   Y. Wen, J. Si, A. Brandt, X. Gao, and H. H. Huang, "Online Reinforcement Learning Control for the Personalization of a Robotic Knee Prosthesis," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2346–2356, Jun. 2020.

[44]   E. J. Scheme, B. S. Hudgins, and K. B. Englehart, "Confidence-based rejection for improved pattern recognition myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 6, pp. 1563–1570, 2013.

[45]   S. Tallam, P. Raghu, D. Macisaac, E. Scheme, and S. Member, "Decision-change Informed Rejection Improves Robustness in Pattern Recognition-based Myoelectric Control Decision-change Informed Rejection Improves Robustness in Pattern," 2023.