Electromyography Based Decoding of Dexterous, In-Hand Manipulation Motions With Temporal Multichannel Vision Transformers

Ricardo V. Godoy, *Graduate Student Member, IEEE*, Anany Dwivedi, *Member, IEEE*, and Minas Liarokapis, *Senior Member, IEEE*

***Abstract*— Electromyography (EMG) signals have been used in designing muscle-machine interfaces (MuMIs) for various applications, ranging from entertainment (EMG controlled games) to human assistance and human augmentation (EMG controlled prostheses and exoskeletons). For this, classical machine learning methods such as Random Forest (RF) models have been used to decode EMG signals. However, these methods depend on several stages of signal pre-processing and extraction of hand-crafted features so as to obtain the desired output. In this work, we propose EMG based frameworks for the decoding of object motions in the execution of dexterous, in-hand manipulation tasks using raw EMG signals input and two novel deep learning (DL) techniques called Temporal Multi-Channel Transformers and Vision Transformers. The results obtained are compared, in terms of accuracy and speed of decoding the motion, with RF-based models and Convolutional Neural Networks as a benchmark. The models are trained for 11 subjects in a motion-object specific and motion-object generic way, using the 10-fold cross-validation procedure. This study shows that the performance of MuMIs can be improved by employing DL-based models with raw myoelectric activations instead of developing DL or classic machine learning models with hand-crafted features.**

***Index Terms*— Electromyography, motion decoding, dexterous manipulation, deep learning, transformers.**

I. Introduction

HUMAN-MACHINE Interfaces (HMI) are finding an increased use in activities of daily living in recent years. For this purpose, biological signals can be used to develop such interfaces, as they carry vital information from the human physiological system. In tasks such as controlling bionic devices, e.g. prosthetic arms and hands, the most commonly employed method is Electromyography (EMG). These signals measure the myoelectric activations of the human muscles generated during contraction and offer an intuitive method for developing HMIs. EMG-based interfaces can decode human movement intention to classify hand gestures and motions [1], [2], as well as the execution of in-hand manipulation motions with an object or continuous human-hand motions [3], [4]. One of the main types of dexterous, in-hand manipulation is Equilibrium Point Manipulation (EPM), in which the contact points of the fingers remain relatively stationary on the object surface while the object is manipulated (see Fig. 1). Robotic arm-hand systems are able to achieve EPM [5], which can be employed to execute tasks such as object inspection or in-hand repositioning or reorientation.

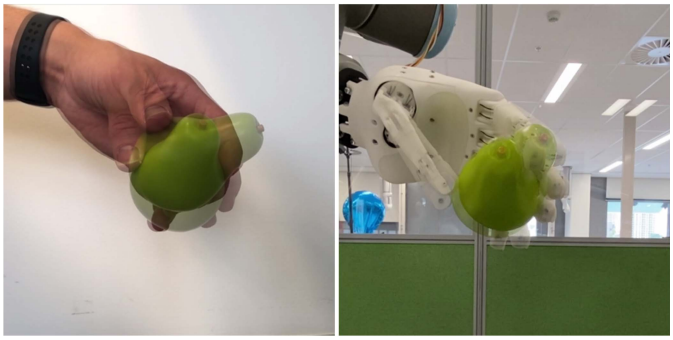

Fig. 1. EMG-based decoding of object motion can be used for dexterous control of robotic or prosthetic arm-hand systems.

Developing an EMG-based control scheme for intuitively executing EPM tasks with a robot or prosthetic hand is a new research direction that has achieved promising results [6], [7]. Machine learning (ML) techniques have been employed to analyse and decode EMG signals in the past few years. A classic machine learning model-based control system for an assistive device generally depends on prior signal pre-processing and feature engineering steps before obtaining the desired classification/regression output [8]. With classic tools such as RF, a feature vector set is extracted from raw data after processing the signal. Time-domain (TD) features have been proved to be a feature class computationally less expensive to calculate, achieving more consistent performance compared to frequency-domain features [9]. Castellini *et al.* [10] compared the results achieved by Neural Networks (NN), Support Vector

Manuscript received 1 April 2022; revised 26 June 2022; accepted 29 July 2022. Date of publication 5 August 2022; date of current version 11 August 2022. *(Corresponding author: Minas Liarokapis.)*

Ricardo V. Godoy and Minas Liarokapis are with the New Dexterity Research Group, Department of Mechanical and Mechatronics Engineering, The University of Auckland, Auckland 1010, New Zealand (e-mail: rdeg264@aucklanduni.ac.nz; minas.liarokapis@auckland.ac.nz).

Anany Dwivedi is with the Chair of Autonomous Systems and Mechatronics, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91052 Erlangen, Germany (e-mail: anany.dwivedi@fau.de).

Digital Object Identifier 10.1109/TNSRE.2022.3196622

Machines (SVM), and Locally Weighted Projection Regression to predict the type of grasp and the grasping force through regression. It was found that none of the tested approaches showed outstanding results among the others, indicating that ML as a whole is a viable approach. Liarokapis *et al.* [11] proposed a task-specific framework for myoelectric activations based on decoding the reach to grasp motions. When comparing the performance of RF with Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis, K-Nearest Neighbours (kNN), NN, and SVM, they found that task-specific models outperform general models, and RF methodology based learned models showed better performance than other learning techniques in classification and estimation accuracy.

Deep learning (DL) approaches have great potential for decoding the human motion or intention from the myoelectric activity. Due to their large amount of parameters compared to conventional function approximators and their non-linear activation functions, DL techniques allow relating more abstract domains and counter domains. End-to-end DL-based models automatically identify and learn high-level features from processed input or raw data using multiple hidden layers, resulting in an increasingly complex and robust system without the need for prior feature extraction. Two well-established DL methods are convolutional neural networks (CNN) and recurrent neural networks (RNN). In recent studies, they have been employed in executing several classification [12] and regression [13]–[15] tasks. In [14], the authors proposed a scheme for estimating the direction and magnitude of the force applied to a grasped object using sEMG of the forearm with a CNN. Chen *et al.* [15] predicted the force of the multi-DOF individual fingers simultaneously based on high-density EMG signals. They compared the performance of a CNN and a CNN plus RNN models with classical methods such as common spatial pattern. Chen *et al.* concluded that methods based on neural networks significantly outperform traditional methods.

Transformer architectures [16], represent the state-of-the-art in Natural Language Processing (NLP) tasks and have recently been employed in new fields, such as image recognition through the Vision Transformer (ViT) [17]. This architecture has great potential in solving problems that hinder the adoption of RNNs and CNNs, such as the inherent impossibility of parallelisation of the former and the significant computational power needed for training the latter. Recent research has developed Transformer-based models for usage in other tasks. Regarding biological signals, Krishna *et al.* [18] proposed an automatic speech recognition model based on Transformer using as input statistical features extracted from EEG signals. Other recent works also employed Transformer-based models in classification tasks using as input EEG, for emotion recognition [19], and EMG signals, for hand gesture classification [20]. These recent advances open up a new range of application areas. However, to the best of the authors' knowledge, no Transformer-based model has ever been developed for regression using raw biological signals' data with multiple channels and time steps as input.

In our previous works, we proposed a learning scheme based on the RF regression method to map the myoelectric activations of the muscles of the forearm and the hand to the object's motion. We studied the optimal muscle selection for the sEMG-based decoding of these in-hand manipulation motions [6], [21]. Then we explored how the EMG signals vary across different subjects of different genders and with different hand sizes, assessing the decoding models' performance [3]. In this paper, we extend our previous works by proposing a new Framework based on two novel Transformer-based regression models. We further compare the results with the results obtained with a CNN benchmark model and with the results of the previously proposed RF model.

TABLE I
SUBJECTS' INFORMATION. *F* STANDS FOR FEMALE, *M* FOR MALE, *R* FOR RIGHT, AND *L* FOR LEFT

Subject	1	2	3	4	5	6	7	8	9	10	11
Gender	F	F	F	F	F	F	M	M	M	M	M
Hand Size (mm)	165	164	170	163	165	169	170	180	200	190	190
Handedness	R	L	R	R	R	R	L	R	R	R	R

The rest of the paper is organised as follows. Section II presents the details regarding the dataset used in this work, a comprehensive description of the Transformers and ViT architectures, and a brief explanation of the CNN benchmark model. Section III presents the results obtained, which are discussed in detail in Section IV. Finally, Section V concludes the paper and presents potential future directions.

II. METHODS

A. Dataset

We used the dataset collected by Dwivedi *et al.* [3] to test the proposed models' performance. The dataset was collected for 11 non-disabled subjects, five males and six females. More information regarding the subjects can be found in Table I. The experiments were performed by each subject with their dominant hand. Each subject performed 3-dimensional equilibrium point manipulation tasks using the Rubik's cube, the chips can from the Yale-CMU-Berkeley (YCB) grasping object set [22], and a custom-made off-center cube. Each manipulation task session was executed with a sequence starting with a 5 sec rest period followed by five repetitions of the manipulation motion for each trial. Adequate time to rest (approximately 30 sec) was given to each subject between trials to reduce the muscles' fatigue. There were 10 of these trials per session. The manipulation tasks performed during the experiments were: pitch, roll, and yaw. More information regarding the manipulation tasks can be found in [3].

The myoelectric activations were measured from eight muscles of the hand and eight forearm muscles using double differential EMG electrodes. The EMG signals were acquired at a sampling rate of 1200 Hz by the bioamplifier, which bandpass filtered the data using a Butterworth filter (5 Hz-500 Hz). The electric line noise was filtered out using a notch filter of 50 Hz.

B. Preprocessing

To evaluate the motion decoding capabilities of our proposed methods, we tested the models for raw and processed data. In order to train models using these methods, the input data needs to be segmented first.

1) Window Size: The procedure described in [3] was employed to segment the data. The signals were segmented into sample sets using a sliding window of 200 ms with a 10 ms increment. According to the literature, the window size is selected to be larger than 125 ms to avoid high biases and variance [23] and smaller than 300 ms due to real-time constraints [24].

2) Raw Data: Raw data implies minimal preprocessing is employed. In the case of raw data, the signals are only filtered by the bioamplifier and segmented before being fed to the algorithms. The use of raw data as input is only possible due to the ability of DL algorithms to learn discriminative features even from noisy data. The RF method can not be used to train successful decoding models, as shown in our last work [3]. Employing automatic feature extraction in other biological signals, such as EEG, using DL has been reported as being more robust and with more potential than those hand-crafted features [25]. Our models identify patterns and characteristics that feature engineering could miss.

3) Processed Data: Three time-domain features were extracted from each EMG channel: Root Mean Square Value (RMS), Waveform Length (WL) and Zero Crossings (ZC). More information regarding the features used can be found in [26].

C. Training and Evaluation

Our models were trained on a Google Colab Pro virtual machine with GPU. The models were developed in Python using Tensorflow and Keras, employing a hyperparameter optimization framework [27] during 200 trials. Then, the hyperparameters were fine-tuned by executing cross-validation with 10% of the available training data for optimization by empirical evaluation. The mean squared error (MSE) loss function was employed during training. The MSE is defined as follows

$$l(y, \hat{y}) = (y - \hat{y})^2 \tag{1}$$

where l is the loss function, y is the desired output, and $\hat{y}$ is the predicted output. All models used Adam as the optimizer [28].

The trained model's efficiency is assessed using the Pearson correlation coefficient and the percentage of the Normalized Mean Square Error (NMSE) representing accuracy in comparing the predicted and the actual object motion. The NMSE value of 0% denotes a bad fit, whereas the NMSE value of 100% denotes that the two trajectories are identical. The NMSE value is derived as follows

$$NMSE(\%) = 100 * \left(1 - \frac{\|x_r - x_p\|^2}{\|x_r - mean(x_r)\|^2}\right) \tag{2}$$

where, $\|.\|$ indicates the 2-norm of a vector, x_r is the actual reference motion, and x_p refers to the predicted motion. All the results presented in Section III are an average of the 10-fold cross-validation, in which one separated repetition of the dataset is used for testing per fold.

To assess the robustness of our algorithms, we compared the results for specific and generalized models, analyzing four different sets:

Subject-Specific and Object-Specific Models: For each subject, we trained and tested one model for each object.

Subject-Specific and Object-Generic Models: For each subject, we trained and tested one model for all the objects.

Subject-Generic and Object-Specific Models: With this set, we trained and tested subject-generic and object-specific models for females, males, small hand size (hand length $\leq$ 165mm), medium hand size (165mm $<$ hand length $\leq$ 185mm), and large hand size (hand length $>$ 185mm).

Subject-Generic and Object-Generic Models: With this set, we trained and tested subject-generic and object-generic models for females, males, small hand size, medium hand size, and large hand size.

Finally, we evaluated the prediction time of each model for raw and processed data as input to assess the applicability of the solutions developed here in online applications.

D. Models

The following sections describe the DL models designed for decoding continuous motion (regression) using EMG activations as input. The last layer in all models comprises three neurons with a linear activation function to perform the roll, pitch, and yaw regression.

1) CNN: The first DL model that we built for regression is a CNN. This technique can identify patterns and extract spatial characteristics of the data. It is one of the most well-established DL techniques, representing the state-of-the-art in several tasks and application fields. Hence, the CNN is evaluated as a benchmark model in order to compare its results with our novel DL techniques. Our CNN model, shown in Fig. 2, comprises three convolutional blocks. Each block contains a convolutional layer, followed by batch normalization [29] and dropout [30] layers. The dropout rate is set to 0.3. The first two blocks also count with max-pooling layers with dimensions of 1×5 each. The filters of the convolutional layers have dimension of 1×20, 4×4, and 2×2, respectively. Three fully-connected layers follow the convolutional blocks with 256, 128, and 64 neurons.

2) TMC-T: The Transformers networks [16] were a milestone in NLP applications, as most of the state-of-the-art algorithms in this area are based on this architecture. Transformers are designed to process sequential data without suffering from vanishing gradients like the RNN, without presenting such complexity as the GRU or LSTM or the impossibility of parallelization inherent to these recurrent techniques. These architectures are based only on attention mechanisms, dispensing with any convolution or recurrence.

Transformer architectures employ attention mechanism to create an attention-based representation for each element in the input sequence. Then, the Transformer focuses on the regions of most significant interest for a given input and,

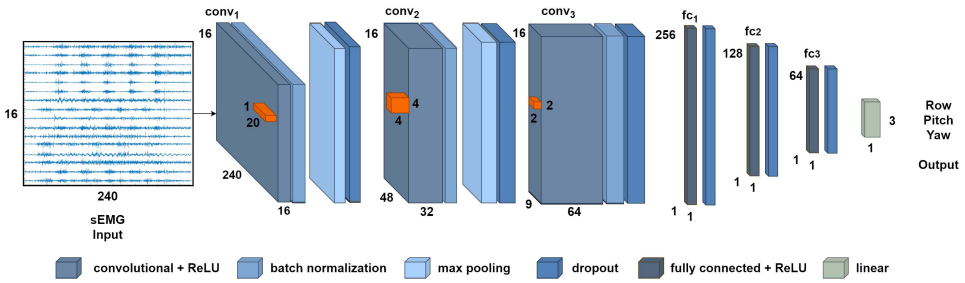

Fig. 2. CNN Model for raw EMG data. The EMG signals are 16×240 matrices, in which the lines are the 16 electrode channels, and the columns are 240 time-steps, i.e. windows of 200 ms acquired at 1,200 Hz. Input dimensions will be gradually reduced through max-pooling layers while the relevant input information is maintained. Filters are shown in the figure in orange. The three time-domain features are extracted and fed to the model when processed data is used as input with dimension of 16×3. The filters' sizes are 4×1, 4×3, and 2×3 respectively and no max-polling is employed.

consequently, spends a greater computational resource in this area. Unlike the attention mechanisms employed with RNNs, the Transformer computes these representations in parallel for each input element. The attention mechanism used by Vaswani *et al.* [16] was the Scaled Dot-Product Attention, given by

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3}$$

where $\sqrt{d_k}$ is the so-called scale factor, and Q, K, and V are vectors called query, key, and value, respectively, that are going to be used inside attention layers in order to compute the attention value for each element.

Vaswani *et al.* [16] employed attention in different positions of different representations of input subspaces through a mechanism called Multi-Head Attention, which allows parallel computation and calculates a richer representation of the input sequence. In the Multi-Head Attention, the same Q, K, and V vectors are multiplied by learned weight matrices. Hence, the attention is calculated for each head h, and the concatenation of these three values is multiplied by a matrix W_O to generate the output of the Multi-Head Attention, as follows

$$\begin{aligned} MultiHead(Q, K, V) &= concat(head_1, \ldots, head_h)W_O \\ head_i &= Attention(Q\ W_i^Q, K\ W_i^K, V\ W_i^V), \end{aligned} \tag{4}$$

where W_i^Q, W_i^K e W_i^V are the learned weight matrices, one for each head. The Transformers' encoder receives the input after going through an embedding to convert each input element to vectors of the same dimension. Following the embedding step, since this model does not use convolution or recurrence, position information for each element is added to the input via a positional encoding. Then, these embeddings get fed to a Multi-Head Attention block within the encoder with h heads. The resulting matrix is provided to a feed-forward network. Residual connection (Add) [31] is employed after both the Multi-Head Attention and the feed-forward network to pass along positional information through the encoder, together with a normalization (Norm) layer [32] to speed up learning. This encoder structure is shown in Fig. 3.

Advantages of using Transformers are the ability to perform parallel computing and fast training time at the cost

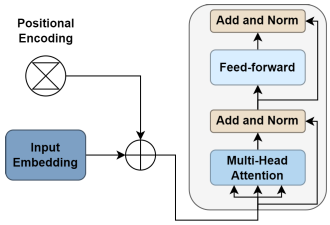

Fig. 3. Transformers' encoder.

of not supporting large input sequences since the attention mechanisms scale quadratically with the input length. For many machine translation applications, in which the input is not that long, the quadratic cost to run the algorithm might not be a problem. However, the quadratic cost represents an obstacle with biological signals acquired at high frequencies for several seconds or even hours. To fulfil the task of processing biological signals with several channels, we developed a Transformer-based model named Temporal Multi-Channel Transformer (TMC-T). The TMC-T model comprises a Transformer block with eight heads and feed-forward networks with 32 neurons. For position encoding, learnable embeddings were used. For token embedding, a convolutional network was used. Using a CNN to generate the inputs' embedding has two purposes:

1) Learn and extract the embeddings. This model employed an embedding dimension of 32 for each token, which is the number of filters in the last convolutional layer within the CNN block.
2) Reduce the input dimension. Since the Transformers scale quadratically with the input length and our input is a matrix of 16×240, i.e. 16 channels of EMG samples of 200 ms acquired at 1,200 Hz, the convolutional layers followed by max-pooling layers reduce the input size while keeping the most relevant information.

For raw data as input, the CNN block is composed of convolution layers followed by batch normalization and a dropout with a rate of 0.3, and max-pooling layers after each of the first two convolution layers. There are three convolution layers of 16, 32, and 32 filters of dimensions 1×20, 4×4, and 2×2. The max-pooling layers have dimensions of 1×5 and 1×4. After the CNN block, the output is reshaped to a

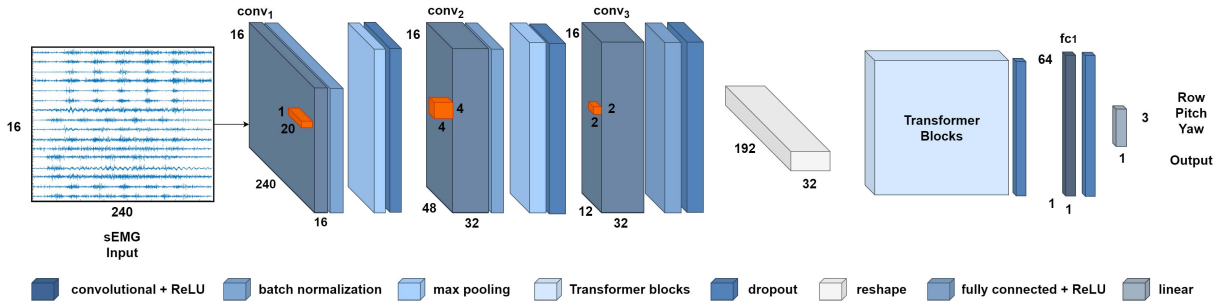

Fig. 4. TMC-T Model for raw EMG data. Three convolution layers extract the embeddings and reduce the input dimension. After that, the result of the convolutions is flattened and supplied to Transformers' blocks. For processed data, filters of 4×1, 4×3, and 2×1 dimensions are employed. A max-pooling layer of 2×1 is used between the second and third convolutional layers, reducing the data dimensions from 16×3 to 8×3. After the convolutional layers, the data is reshaped to 24×32, in which 32 is the embedding dimension.

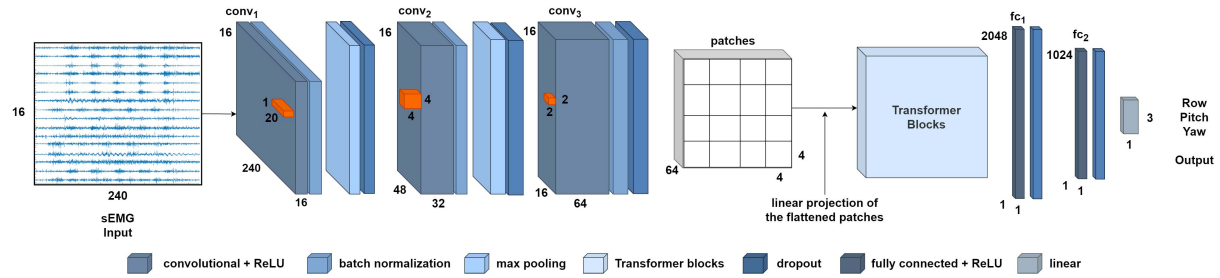

Fig. 5. TMC-ViT Model for raw EMG data. Three convolution layers extract the embeddings and reduce the input dimension to a 2D data grid of 16×16 with an embedding size of 64, which are interpreted as images by ViT and further split into patches of 4×4. When processed data is used as input, only one max-pooling layer of 2×1 is employed between the second and third convolution. The number of filters is 16, 32, and 32 with dimensions of 4×1, 4×3, and 2×1 respectively. For this case, the embedding dimension is 32. The patches for processed data have dimensions of 4×3.

matrix of dimensions 192×32, i.e. an input sequence of length 192 and embedding size of 32. After the Transformer blocks, a dropout of 0.5 and a dense layer of 64 neurons with ReLU activation function is employed (see Fig. 4).

3) TMC-ViT: ViT is a Transformer model adapted to use images as input. Thus, instead of processing 1D sequential data, ViT will use 2D images as input. In a first step, the ViT will subdivide the input image $x \in \mathbb{R}^{H \times W \times C}$ into a sequence of flattened 2D patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (H, W) is the resolution of the original image, C is the number of channels, (P, P) is the resolution of each image patch, and $N = HW/P^2$ is the resulting number of patches. Then, a linear embedding sequence of these patches and position embeddings are provided as input to a Transformer encoder (see Fig. 3). While the position embedding adds input topology information, the ViT processes the image with a linear projection of the flattened patches, whose components indicate low-dimensional correlations in the patches, and the Multi-Head Attention mechanism aggregates image information across all layers. Dosovitskiy *et al.* [17] employed this architecture for classifying 16×16 images. To adapt this network to process multi-channel EMG signals as input, we developed the Temporal Multi-Channel Vision Transformer (TMC-ViT) model. A CNN was used at the input to create the embeddings and reduce the input matrix to dimensions 16×16. The patches we used have the size of 4×4. Thus, sequential signals from multiple channels will be interpreted as 2D images. The CNN mentioned above has 16, 32, and 64 filters of dimensions 1×20, 4×4, and 2×2. The first two convolutional layers are followed by max-polling layers of 1×5 and 1×3, respectively. After each convolutional layer, a dropout rate of 0.2 and batch normalization layers are employed. Again, learnable embeddings were used for the position encoding, and a convolutional network was used for the tokens embedding. Since the last layer of the convolution has 64 filters, embedding has a dimension of 64. The number of attention heads and Transformer layers adopted was respectively 4 and 8. The last dense layers have dimensions 2,048 and 1,024. This model is illustrated in Fig. 5.

III. Results

This section presents the object motion decoding accuracies obtained using the TMC-ViT, TMC-T, and CNN. Respective decoding models were developed using raw and processed myoelectric activations. All the results presented are an average of 10-fold cross-validation. For each evaluation set, the results for the motion decoding models developed using raw data will be presented first, followed by the results for the models developed using the processed data. The DL models were statistically validated using the analysis of variance (ANOVA). The null hypothesis for the analysis was that all models are the same. However, a p-value of 0.0049 was obtained for the models developed using raw EMG data, implying the results are statistically significant (p-value <0.05), thus rejecting our null hypothesis and concluding that there is a significant difference among the tested models developed

TABLE II
SUBJECT-SPECIFIC AND OBJECT-SPECIFIC MODELS FOR RAW DATA. *C* STANDS FOR CORRELATION AND *A* FOR ACCURACY

Model	CNN						TMC-T						TMC-ViT					
Obj.	Rubiks		Chips		Off-center		Rubiks		Chips		Off-center		Rubiks		Chips		Off-center	
Subj.	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A	C	A
1	88.89	76.70	74.14	47.79	63.96	42.64	92.23	81.82	83.69	60.46	59.89	51.05	92.42	82.25	80.70	56.69	81.19	58.76
2	83.55	65.83	81.87	60.78	77.36	53.65	87.57	73.67	81.01	58.77	81.13	60.23	89.51	78.58	84.32	66.99	82.32	63.03
3	88.85	76.17	71.72	37.79	77.68	49.46	90.90	81.19	64.09	40.57	84.64	67.99	92.47	83.42	81.26	53.38	85.41	68.65
4	61.08	26.85	77.58	57.59	65.81	37.78	70.79	39.03	83.59	59.90	71.96	44.11	69.65	40.49	86.53	64.71	68.31	34.31
5	82.63	64.74	69.07	40.36	79.86	54.59	87.31	70.23	73.52	45.68	83.40	62.33	89.03	75.38	76.48	54.35	83.69	62.05
6	86.54	71.86	82.56	62.15	76.73	52.05	87.82	73.36	84.70	64.15	75.50	47.57	88.27	73.64	84.19	63.88	85.07	67.44
7	95.76	90.52	86.07	71.09	91.98	82.72	95.98	90.95	86.08	70.89	93.97	86.93	96.64	92.08	88.80	75.49	94.26	87.13
8	89.10	78.38	88.43	74.89	90.95	81.73	90.67	80.89	88.78	75.20	92.18	83.83	91.62	83.00	89.24	77.04	93.12	85.90
9	90.11	80.01	79.09	58.10	80.18	57.50	90.97	80.97	84.30	66.88	86.81	70.25	91.42	82.28	85.31	70.66	87.67	72.04
10	85.94	72.55	78.84	58.17	85.43	71.32	88.78	77.66	88.78	77.66	88.79	76.68	88.70	77.64	82.48	62.95	88.24	76.45
11	88.89	76.70	78.45	57.61	82.70	65.09	85.14	72.30	81.69	61.99	85.16	67.57	85.94	73.03	81.29	61.76	84.80	68.05
AVG	85.58	70.94	78.89	56.94	79.33	58.96	88.02	74.73	81.84	62.01	82.13	65.32	88.70	76.53	83.69	64.35	84.92	67.62
STD	8.85	16.21	5.81	11.37	8.82	14.75	6.41	13.21	7.21	11.27	9.83	14.06	6.92	13.13	3.77	7.87	6.87	14.33

for raw EMG data. On the contrary, for the models developed using processed EMG data, a p-value of 0.40 was obtained, indicating that there is no significant difference when trained using the processed EMG data, due to the limited amount of information that can be extracted from processed data.

A. Subject-Specific and Object-Specific

In this set, one model was trained for each object in a subject-specific way.

1) Raw Data: The results obtained by our Subject-Specific and Object-Specific models for raw EMG data are shown in Table II.

2) Processed: The results obtained by our Subject-Specific and Object-Specific models for processed data are shown in Table III. The results are further compared with our previously RF model [3].

B. Subject-Specific and Object-Generic

Here, we developed subject-specific models for all objects.

1) Raw Data: The results obtained for the subject-specific and object-generic models for raw data are presented in Table IV. It can be noticed that our TMC-ViT model surpassed the CNN benchmark model in both correlation and accuracy, achieving 89.68% and 79.09%, respectively. Moreover, the TMC-ViT presented a correlation above 80% for all tested subjects, demonstrating its robustness in learning the unique characteristics of each individual's EMG, performing the regression with more than 60% accuracy for all subjects, reaching the regression up to 93.63% accuracy for subject number nine. The TMC-T model achieved better accuracy and competitive correlation compared to the CNN model.

2) Processed Data: The results obtained for the subject-specific and object-generic models for processed data are presented in Table V together with the results achieved by the RF model. All the DL techniques learned from the features extracted during preprocessing. The DL models trained in this work and the RF model from our previous paper [3] achieved similar results. This indicates that our models could have reached a threshold in which the algorithms learned as much as possible from the data available. The TMC-ViT achieved the best results, with an average correlation of 81.01% and an accuracy of 63.10%. As was expected, the RF model could benefit from processed data, presenting competitive performance with the DL models when features extracted through feature engineering were used as input.

TABLE III
SUBJECT-SPECIFIC AND OBJECT-SPECIFIC MODELS FOR PROCESSED DATA

Model	CNN						TMC-T					
Obj.	Rubiks		Chips		Off-center		Rubiks		Chips		Off-center	
Subj.	C	A	C	A	C	A	C	A	C	A	C	A
1	87.31	73.91	73.68	51.70	89.90	78.59	85.22	69.04	73.59	52.28	84.86	70.23
2	88.87	78.06	75.73	55.71	76.70	58.43	88.19	76.65	75.17	54.73	77.16	58.42
3	83.83	67.96	74.20	60.74	82.58	65.62	83.81	68.96	74.19	60.22	80.37	61.67
4	62.49	31.49	71.09	48.03	62.38	36.16	64.17	34.37	74.43	53.35	65.70	38.96
5	87.25	71.20	64.55	42.76	77.98	58.59	90.08	78.91	67.81	50.81	81.19	62.50
6	83.32	62.58	80.08	61.13	87.78	73.80	83.36	61.66	78.57	61.53	86.03	71.60
7	90.10	79.14	85.28	70.10	91.31	81.36	89.63	77.03	84.91	69.76	90.02	78.99
8	85.82	71.20	84.91	70.21	83.07	65.46	86.72	73.74	84.00	69.46	84.00	69.46
9	89.57	78.85	75.44	55.76	77.37	59.17	88.85	78.11	76.83	59.86	78.97	61.66
10	79.87	61.93	77.69	54.96	77.87	57.82	79.85	62.55	78.64	59.08	76.47	57.25
11	87.66	73.07	81.67	63.57	83.19	67.82	86.30	70.89	85.53	71.44	84.86	70.23
AVG	84.19	68.13	76.76	57.70	80.92	63.89	84.20	68.36	77.61	60.23	80.88	63.72
STD	7.81	13.50	6.12	8.57	8.00	12.36	7.31	12.73	5.47	7.31	6.47	10.52

Model	TMC-ViT						RF					
Obj.	Rubiks		Chips		Off-center		Rubiks		Chips		Off-center	
Subj.	C	A	C	A	C	A	C	A	C	A	C	A
1	85.84	69.98	86.49	73.55	86.49	73.55	92.6	83.61	76	52.4	75.67	59.6
2	88.72	77.83	73.62	52.29	77.26	58.31	82.62	67.91	76.72	55.68	75.55	55.02
3	84.17	69.36	72.54	56.89	80.02	60.70	87.06	74.26	79.77	60.41	86.6	72.15
4	64.99	34.70	74.38	53.28	65.77	40.51	61.64	32.49	74.24	49.32	53.45	27.01
5	90.43	79.77	67.70	50.77	81.34	62.71	87.86	75.01	77.37	54.95	77.84	56.72
6	83.78	62.38	78.71	61.89	86.45	71.91	79.64	57.93	76.15	53.64	88.86	66.65
7	89.59	76.92	85.11	70.70	91.98	82.72	89.61	77.91	78.6	59.72	88.49	75.09
8	86.62	73.87	84.45	70.00	90.95	81.73	85.78	72.18	83.13	67.35	86.98	73.11
9	88.45	77.32	76.53	59.55	78.91	61.50	87.27	73.64	81.44	63.64	85.15	69.38
10	79.94	62.54	78.35	59.27	76.92	58.45	78.36	57.88	70.97	45.99	73.9	60.14
11	86.24	71.35	86.02	72.22	82.34	66.07	82.89	67.71	78.98	58.61	83.64	68.09
AVG	84.43	68.73	78.54	61.86	81.68	65.29	83.21	67.32	77.58	56.52	79.65	62.09
STD	7.11	12.72	6.30	8.44	7.38	11.99	8.31	13.93	3.37	6.24	10.33	13.49

C. Subject-Generic and Object-Specific

This section presents the results for models trained for a set of subjects and each object.

TABLE IV
SUBJECT-SPECIFIC AND OBJECT-GENERIC MODELS FOR RAW DATA

Model	CNN		TMC-T		TMC-ViT	
Subj.	C (%)	A (%)	C (%)	A (%)	C (%)	A (%)
1	91.00	81.64	85.41	72.95	93.16	85.00
2	70.22	46.71	79.35	69.09	80.89	62.02
3	79.00	57.98	84.46	69.72	86.14	70.77
4	83.94	68.47	84.81	70.64	87.88	75.54
5	91.12	82.42	78.87	60.51	94.77	88.80
6	80.64	62.4	79.58	60.54	82.74	64.71
7	94.41	88.56	94.80	89.23	95.55	90.75
8	90.95	82.40	91.27	82.83	92.32	84.75
9	94.57	88.97	93.77	86.74	97.08	93.6
10	84.92	71.44	86.82	74.68	87.01	74.97
11	86.23	74.39	86.97	74.58	89.03	78.78
AVG	86.09	73.22	86.01	73.77	89.68	79.09

TABLE V
SUBJECT-SPECIFIC AND OBJECT-GENERIC MODELS FOR PROCESSED DATA

Model	CNN		TMC-T		TMC-ViT		RF [3]	
Subj.	C (%)	A (%)	C (%)	A (%)	C (%)	A (%)	C (%)	A (%)
1	74.72	54.05	72.43	51.68	74.45	53.69	73.37	56.50
2	80.83	64.95	79.83	63.03	81.56	65.977	79.00	61.03
3	81.07	65.46	80.95	65.07	81.14	65.79	84.24	69.39
4	63.97	37.32	64.69	39.34	64.78	39.28	62.70	36.82
5	73.32	48.78	77.16	58.83	76.36	56.69	79.96	61.41
6	83.69	66.11	82.29	64.60	83.86	66.56	80.14	61.75
7	89.28	78.58	88.43	77.37	89.68	79.71	87.05	73.82
8	84.77	70.51	84.80	72.03	85.58	72.09	85.76	72.48
9	79.34	62.13	79.56	62.92	81.38	65.64	82.41	64.82
10	75.42	55.74	75.45	56.38	76.46	57.94	76.16	55.77
11	84.50	69.77	82.51	67.09	84.85	70.76	81.29	65.32
AVG	79.17	61.22	78.92	61.67	80.01	63.10	79.37	61.74

1) Raw Data: The results obtained for subject-generic and object-specific models for raw data are shown in Table VI. The TMC-ViT model achieved the best results compared with the other models for raw data for any group of subjects or objects. The TMC-T and CNN obtained competitive results with each other.

2) Processed: The results obtained for subject-generic and object-specific models for processed data are shown in Table VII. Once again, all the models for processed data presented similar results. The DL models achieved better performance for the males, those with medium and larger hand sizes.

D. Subject-Generic and Object-Generic

1) Raw Data: The results obtained for subject-generic and object-generic models for raw data are shown in Table IX.

TABLE VI
SUBJECT-GENERIC AND OBJECT-SPECIFIC MODELS FOR RAW DATA

Object	Rubik's Cube		Chips Can		Off-Center Mass Cube	
Group	C (%)	A (%)	C (%)	A (%)	C (%)	A (%)
TMC-ViT						
Female	94.95	89.70	75.91	55.91	92.72	85.48
Male	91.49	83.60	92.92	86.09	95.28	90.55
Small hand	93.48	86.88	80.43	93.31	93.76	87.15
Medium hand	93.38	86.64	85.13	71.15	94.00	87.86
Large hand	88.81	78.66	91.84	83.88	95.47	90.78
TMC-T						
Female	91.24	84.50	73.11	53.54	84.85	72.28
Male	89.88	80.88	90.54	80.74	94.28	88.44
Small hand	88.22	77.44	79.26	63.33	83.06	70.02
Medium hand	91.78	83.69	82.47	66.09	91.43	82.63
Large hand	87.42	76.47	88.98	77.35	94.88	89.30
CNN						
Female	93.87	87.60	72.37	52.84	82.92	66.03
Male	89.78	80.70	91.30	83.34	94.31	88.76
Small hand	92.09	84.15	78.55	62.06	87.15	74.09
Medium hand	91.91	84.06	81.76	65.73	90.27	81.13
Large hand	86.93	75.63	89.60	80.23	94.54	89.03

The TMC-ViT achieved better performance than the TMC-T and CNN models, presenting correlation above 81% and accuracy above 65% for all the tested groups. The males achieved 93.14% correlation and 86.6% accuracy, representing the group with the highest results. Fig. 6 shows the actual vs decoded motion for the participants of the male group achieved by the TMC-ViT model.

2) Processed: The results obtained for subject-generic and object-generic models for processed data are shown in Table VIII. The TMC-ViT model achieved the best results, whereas the others showed similar results. The medium and large hand sizes presented competitive results for processed data and Transformer-based methods. The CNN and RF models performed worse for the large hand sizes when compared to medium hand sizes, indicating higher robustness of the TMC-ViT and TMC-T models. The DL models performed better when raw data was used as input compared with the RF model.

E. Prediction Time

In this section, we measured the time required by each model to predict a new motion sample. The TMC-ViT and other models were developed and optimized to present a high correlation and accuracy. The time presented here is an average of 100 trials for classifying 12,000 samples. First, we calculated the time required for processing the data, i.e. extracting the three hand-extracted features. It was found that the feature engineering takes an average of 1.43 seconds to extract the features from 12,000 samples. Table X presents the prediction

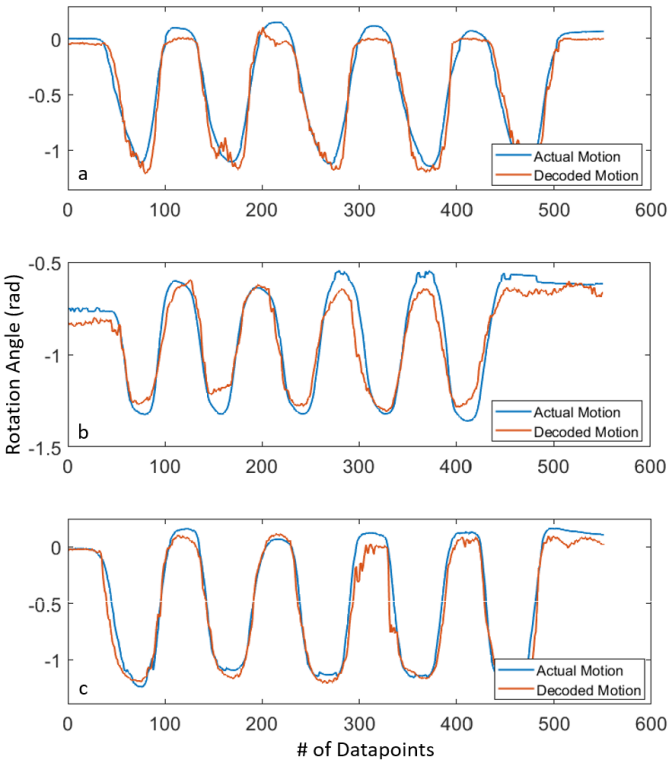

Fig. 6. Plots of actual vs decoded motion from the myoelectric activation of the male participants by the TMC-ViT subject-generic object-generic model. Subfigure (a) presents the pitch motion, subfigure (b) depicts the roll motion, and subfigure (c) shows the yaw motion.

time for the three DL models for raw and processed data as input. In the latter case, the data processing time is already considered. Moreover, in this table, we show each model's prediction frequency and number of parameters. Finally, the results obtained for the RF model are also presented.

For the sake of comparison, we also optimized a deep CNN (DCNN) with a similar number of parameters to the ViT. The DCNN was trained and tested for the subject-generic object-generic set. This subset of experiments was chosen because it is the largest and most generic set with the greatest potential to benefit from a deeper model. The DCNN is composed of three convolutional blocks. The blocks contain two, three, and four convolutional layers with 32, 64, and 128 filters, respectively. This model achieved an accuracy of [49.80%, 84.20%, 47.92%, 75.05%, 82.24%] for the "female", "male", "small hand", "medium hand", and "large hand" groups respectively. Comparing these results with those presented in Table IX shows that the DCNN achieved the worst results among the tested models. Adding complexity to the model did not improve the performance compared to the smaller CNN.

IV. Discussion

A. Subject-Specific and Object-Specific Models

The analysis of the Table II and III highlights the better performance of the DL techniques to the classic ML algorithm tested, i.e. the RF model. All tested DL models outperformed the RF model in correlation and accuracy for raw and processed data. Regarding the tested DL techniques, the success of the Transformer-based models developed here is noticed. Both TMC-ViT and TMC-T outperformed the CNN benchmark model in correlation and accuracy for each data type. The TMC-ViT model achieved more than 83% correlation and 64% accuracy for all objects. Using raw data as input for our DL models enhanced correlation and accuracy. DL algorithms employing raw data showed better results in relation to DL or classical ML methods employing processed data. Higher performance for raw data as input

TABLE VII
SUBJECT-GENERIC AND OBJECT-SPECIFIC MODELS FOR PROCESSED DATA

Object	Rubik's Cube		Chips Can		Off-Center Mass Cube	
Group	**C (%)**	**A (%)**	**C (%)**	**A (%)**	**C (%)**	**A (%)**
TMC-ViT						
Female	78.23	60.93	73.47	53.42	81.01	63.62
Male	83.27	69.21	80.24	63.16	81.64	64.70
Small hand	79.70	63.07	72.59	51.82	80.67	63.47
Medium hand	85.04	71.44	82.48	67.20	86.22	73.04
Large hand	85.12	71.83	82.86	67.40	86.01	72.72
TMC-T						
Female	77.38	58.17	70.20	49.13	76.70	57.43
Male	82.23	67.13	78.12	60.59	79.17	62.15
Small hand	76.77	58.01	70.24	48.83	74.45	54.24
Medium hand	83.31	68.51	80.58	64.41	84.80	71.33
Large hand	83.70	68.79	80.55	64.49	85.33	72.22
CNN						
Female	78.14	60.75	71.99	51.70	80.73	63.51
Male	83.20	68.46	79.36	61.09	81.29	64.91
Small hand	78.26	60.99	70.99	49.91	79.56	61.43
Medium hand	85.15	71.73	81.54	64.70	86.50	73.73
Large hand	81.93	65.79	70.99	49.91	77.11	58.06
RF						
Female	79.87	62.26	75.95	56.07	71.26	52.66
Male	81.86	65.65	76.99	57.06	83.37	67.52
Small hand	79.76	61.92	75.12	54.09	68.68	49.52
Medium hand	84.62	70.40	79.99	62.02	85.22	70.80
Large hand	79.85	62.11	73.93	51.94	80.32	62.31

TABLE VIII
SUBJECT-GENERIC AND OBJECT-SPECIFIC MODELS FOR PROCESSED DATA

Model	TMC-ViT		TMC-T		CNN		RF	
Group	**C (%)**	**A (%)**	**C (%)**	**A (%)**	**C (%)**	**A (%)**	**C (%)**	**A (%)**
Female	74.03	54.05	71.19	50.35	73.73	54.00	72.58	51.86
Male	81.55	63.31	78.71	62.05	80.54	64.58	80.70	63.06
Small hand	72.58	52.16	59.33	47.71	71.33	50.27	70.30	49.31
Medium hand	84.68	71.19	83.25	69.33	84.93	72.01	83.22	67.58
Large hand	84.97	72.04	83.13	68.93	77.75	59.35	77.31	57.76

TABLE IX
SUBJECT-GENERIC AND OBJECT-GENERIC MODELS FOR RAW DATA

Model	TMC-ViT		TMC-T		CNN	
Group	C (%)	A (%)	C (%)	A (%)	C (%)	A (%)
Female	81.41	65.44	79.25	61.16	78.21	60.91
Male	93.14	86.60	89.07	79.00	91.47	83.71
Small hand	84.70	70.71	80.86	62.08	79.90	62.44
Medium hand	88.64	77.09	76.06	62.61	80.13	63.42
Large hand	92.13	84.67	85.86	73.56	89.74	80.60

TABLE X
PREDICTION TIME FOR EACH MODEL

Model	Data type	Prediction time (12000 samples) [sec]	Frequency [kHz]	No. of parameters
TMC-ViT	Raw	2.91	4.12	4,950,067
TMC-T	Raw	1.61	7.44	57,267
CNN	Raw	1.13	10.64	2,418,419
DCNN	Raw	3.08	3.90	5,178,883
TMC-ViT	Processed	3.30	3.64	2,428,531
TMC-T	Processed	2.19	5.47	51,635
CNN	Processed	2.07	5.81	849,331
RF	Processed	1.66	7.22	-

is only achievable due to the ability of DL techniques to learn the relevant features even from raw data, extracting information that could be lost during feature engineering. Another advantage is that using raw data minimizes the need for prior knowledge regarding the signals.

B. Subject-Specific and Object-Generic

The analysis of the Tables IV and V leads to two conclusions: i) once again, employing raw data to train the DL models improved both accuracy and correlation when compared to DL or classic ML techniques for processed data, ii) our TMC-ViT for raw data and processed data outperformed any other model for the respective data type. When comparing the results achieved by the TMC-ViT model for raw data and the RF model from our previous work, we can observe an increase of 10.31% in correlation and 17.35% in accuracy. Another interesting finding is that the performance difference between DL and classical ML techniques is even more significant for the Subject-Specific and Object-Generic models. The Subject-Specific and Object-Generic model's accuracy employing TMC-ViT and raw data is 1.28 times larger than the model's accuracy using RF. For the Subject-Specific and Object-Specific models, for the Chips Can object, for example, this ratio is only 1.14. This fact is explained by DL models outperforming classical ML models the larger the dataset.

C. Subject-Generic and Object-Specific

From the Table VI, it can be noted that the female subjects and those with small or medium hand sizes have a considerable drop in motion decoding accuracy and correlation for the chips can as compared to the Rubik's and the off-center mass

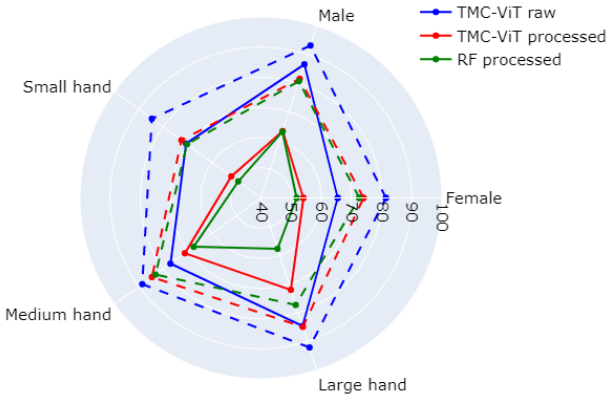

Fig. 7. Average accuracy (solid line) and correlation (dashed line) obtained by the subject-generic object-generic TMC-ViT models for each input data type and DL technique. The results of the RF model for processed data are also presented.

cube. Whereas the male subjects and those with bigger hand sizes have better performance with the off-center mass cube and worse with the chips can. When comparing the results obtained by the subject-generic and object-specific (Table VI) with the subject-specific and object-generic (Table II) models, it is noted that the DL models with raw data could benefit from the larger dataset, performing better for subject-generic models than for the subject-specific models. One thing that is interesting to notice is that, for the RF model, the females and those with smaller hand sizes have a considerable drop in motion decoding accuracy for the off-center mass cube compared to the Rubik's cube, the opposite of the DL models. The DL models performed better when raw data was used as input, surpassing the RF model as expected.

D. Subject-Generic and Object-Generic

In Fig. 7 we present both the correlation (dashed line) and accuracy (solid line) obtained by the TMC-ViT model for raw and processed data. The results of the RF model for processed data are also compared. Here is noticed the behaviour shown in all the training sets: i) the TMC-ViT model achieves higher results than the RF model and ii) using raw data as input enhanced both correlation and accuracy for the DL models.

E. Prediction Time

The DL models performed better for raw data as input than for processed data, showing that removing feature engineering steps during data preprocessing can improve the applicability of the DL models in real-time applications. The DCNN achieved the worst prediction time among the tested models. The TMC-T and the CNN models for raw data presented a shorter prediction time than any other model for processed data, including the RF model. The TMC-ViT is the most robust model, which showed better accuracy and correlation results in all tests. The TMC-ViT is also one of the deepest models, presenting 4,950,067 parameters for raw data and, consequently, a longer prediction time. Even though the TMC-ViT model has shown a longer prediction time than the other models,

it is still a suitable candidate method for online applications with a prediction frequency for 12,000 samples higher than 4 kHz.

V. Conclusion

In this work, we have proposed a novel end-to-end deep learning approach for decoding object motion in dexterous, in-hand manipulation tasks based on EMG signals. The proposed framework employs a Transformer-based architecture modified to receive as input EMG signals in order to achieve motion decoding. In particular, two new models called Temporal Multi-Channel Transformer and Temporal Multi-Channel Vision Transformer are introduced for solving the EMG-based decoding problem. We tested our models with raw and processed data as input and compared the results with a CNN benchmark model and an RF model proposed in previous works, representing the classic machine learning techniques.

Our models have been trained in subject-generic and subject-specific ways and an object-generic and object-specific manner. It can be seen that both the accuracies and the correlations increase when using DL models with raw data instead of DL or classic ML techniques with processed data. The DL models also generalized better than the classic ML models, achieving better results for the subject-generic object-generic model. In terms of accuracy and correlation, the Temporal Multi-Channel Vision Transformer achieved the best results among the tested models. The DL models showed a faster prediction time for raw data than for processed data. Hence, end-to-end DL approaches surpassed the use of processed data and/or classic ML techniques, such as RF.

Future work will focus on the information learned by the DL techniques by evaluating the patterns learned by the different attention heads in the Temporal Multi-Channel Vision Transformer model, using both sEMG and high dimensional-EMG signals as input.

References

[1] M.-F. Lucas, A. Gaufriau, S. Pascual, C. Doncarli, and D. Farina, "Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization," *Biomed. Signal Process. Control*, vol. 3, no. 2, pp. 169–174, 2008.

[2] A. Dwivedi, G. Gorjup, Y. Kwon, and M. Liarokapis, "Combining electromyography and fiducial marker based tracking for intuitive telemanipulation with a robot arm hand system," in *Proc. 28th IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, Oct. 2019, pp. 1–6.

[3] A. Dwivedi, Y. Kwon, A. McDaid, and M. Liarokapis, "A learning scheme for EMG based decoding of dexterous, in-hand manipulation motions," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 10, pp. 2205–2215, Aug. 2019.

[4] M. Simão, N. Mendes, O. Gibaru, and P. Neto, "A review on electromyography decoding and pattern recognition for human-machine interaction," *IEEE Access*, vol. 7, pp. 39564–39582, 2019.

[5] K. Kiguchi and Y. Hayashi, "An EMG-based control for an upper-limb power-assist exoskeleton robot," *IEEE Trans. Syst., Man, Cybern., B (Cybernetics)*, vol. 42, no. 4, pp. 1064–1071, Aug. 2012.

[6] A. Dwivedi, Y. Kwon, A. J. McDaid, and M. Liarokapis, "EMG based decoding of object motion in dexterous, in-hand manipulation tasks," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechtron. (Biorob)*, Aug. 2018, pp. 1025–1031.

[7] A. Dwivedi, J. Lara, L. K. Cheng, N. Paskaranandavadivel, and M. Liarokapis, "High-density electromyography based control of robotic devices: On the execution of dexterous manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3825–3831.

[8] N. Nazmi, M. A. A. Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of EMG signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, p. 1304, 2016.

[9] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Exp. Syst. Appl.*, vol. 40, no. 12, pp. 4832–4840, Sep. 2013.

[10] C. Castellini, P. van der Smagt, G. Sandini, and G. Hirzinger, "Surface EMG for force control of mechanical hands," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 725–730.

[11] M. V. Liarokapis, P. K. Artemiadis, K. J. Kyriakopoulos, and E. S. Manolakos, "A learning scheme for reach to grasp movements: On EMG-based interfaces using task specific motion decoding models," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 5, pp. 915–921, Sep. 2013.

[12] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.

[13] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5974–5983.

[14] Y. Ban, "Estimating the direction of force applied to the grasped object using the surface EMG," in *Haptics: Science, Technology, and Applications*, D. Prattichizzo, H. Shinoda, H. Z. Tan, E. Ruffaldi, and A. Frisoli, Eds. Cham, Switzerland: Springer, 2018.

[15] Y. Chen, C. Dai, and W. Chen, "Cross-comparison of EMG-to-force methods for multi-DoF finger force prediction using one-DoF training," *IEEE Access*, vol. 8, pp. 13958–13968, 2020.

[16] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[17] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[18] G. Krishna, C. Tran, M. Carnahan, and A. H. Tewfik, "EEG based continuous speech recognition using transformers," 2019, *arXiv:2001.00501*.

[19] J. Liu, L. Zhang, H. Wu, and H. Zhao, "Transformers for EEG emotion recognition," 2021, *arXiv:2110.06553*.

[20] A. Burrello *et al.*, "Bioformers: Embedding transformers for ultra-low power sEMG-based gesture recognition," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2022, pp. 1443–1448.

[21] Y. Kwon, A. Dwivedi, A. J. McDaid, and M. Liarokapis, "On muscle selection for EMG based decoding of dexterous, in-hand manipulation motions," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 1672–1675.

[22] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 36–52, Sep. 2015.

[23] M. A. Oskoei and H. Hu, "Myoelectric control systems—A survey," *Biomed. Signal Process. Control*, vol. 2, no. 4, pp. 275–294, 2007.

[24] K. Englehart, B. Hudgin, and P. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 3, pp. 302–311, Mar. 2001.

[25] A. Antoniades, L. Spyrou, C. C. Took, and S. Sanei, "Deep learning for epileptic intracranial EEG data," in *Proc. IEEE 26th Int. Workshop Mach. Learn. for Signal Process. (MLSP)*, Sep. 2016.

[26] A. Turner, D. Shieff, A. Dwivedi, and M. Liarokapis, "Comparing machine learning methods and feature extraction techniques for the EMG based decoding of human intention," in *Proc. 43rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Nov. 2021, pp. 4738–4743.

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019.

[28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2015, pp. 448–456.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[32] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.