Continuous Estimation of Hand Kinematics from Electromyographic Signals based on Power-and Time-Efficient Transformer Deep Learning Network

Chuang Lin*, Chunxiao Zhao, Jianhua Zhang, Chen Chen, Ning Jiang, *Senior Member, IEEE*, Dario Farina, *Fellow, IEEE*, and Weiyu Guo*

***Abstract*—Surface Electromyographic (sEMG) signals contain motor-related information and therefore can be used for human-machine interaction (HMI). Deep learning plays an important role in extracting motor-related information from sEMG signals. However, most studies prioritize model accuracy without sufficient consideration of model efficiency, including the model size, power consumption, and the computational speed of the model. This leads to impractical power consumption, heat dissipation levels and processing time in wearable computation scenarios. Here, we propose an efficient Transformer method that employs the EMSA (Efficient Multiple Self-Attention) and pruning mechanism to improve efficiency and accuracy concurrently, when estimating finger joint angles from sEMG signals. The proposed method does not only achieve state-of-the-art accuracy but can also be deployed on wearable devices to satisfy real-time applications. We applied the proposed model on the Ninapro DB2-dataset to estimate finger joint angles during grasping tasks. RNN series models, Convolution series models, and Transformer series models were used as reference models for comparison. In addition to common model accuracy, the deployment performance of the models was tested on microprocessors, such as Intel CPU i5, Apple M1, and Raspberry Pi 4B. When tested on 38 subjects of the Ninapro DB2, the proposed model resulted in a correlation coefficient of 0.82 ± 0.04, root mean squared error (RMSE) of 10.77 ± 1.48, and normalized RMSE of 0.11 ± 0.01, which were all similar to the results achieved by the state-of-the-art (SOTA) reference methods. Further, the computational time of the proposed methods was 65.99 ms on the Raspberry Pi 4B, which outperformed all the RNN series models and the Transformer series models. The model size and the power (the minimum size and power are 0.39 MB and 2.28 w) consumption of the proposed model also outperformed that of all reference Transformer methods. These experimental results indicate that our model can maintain the accuracy of the SOTA methods while significantly improving efficiency, thus being a promising approach for real-life applications in wearable devices.**

***Index Terms*—sEMG, Continuous Estimation, Finger Kinematics, Transformer, Model Efficiency.**

Manuscript received November 15, 2023; revised December 26, 2023.

Chuang Lin, Chunxiao Zhao, and Jianhua Zhang are with the School of Information Science and Technology, Dalian Maritime Universtiy, Dalian 116026, China (e-mail: chuang.lin@dlmu.edu.cn, 316088072zcx@dlmu.edu.cn, jianhuazhang@dlmu.edu.cn)

Chen Chen is with the State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China (e-mail: cedricchen@sjtu.edu.cn).

Ning Jiang is with the National Clinical Research Center for Geriatrics, Sichuan University West China Hospital; Med-X Center for Manufacturing, Sichuan University, Sichuan 610017, China (e-mail: jiangning21@wchscu.cn).

Dario Farina is with the Department of Bioengineering, Imperial College London, SW7 2BX London, U.K. (e-mail: d.farina@imperial.ac.uk).

Weiyu Guo is with the Artificial Intelligence Thrust, Information Hub, The Hong Kong University of Science and Technology, Guangzhou 511466, China, and also with the Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou 511466, China (e-mail: guoweiyu@126.com).

The corresponding authors are Chuang Lin and Weiyu Guo

Source code is available online at: https://github.com/zcx-1999/sEMG_Transformer_ESMA

I. Introduction

With the development of artificial intelligence, neural networks have been gradually exploited in the field of electromyography processing. Specifically, deep neural networks (DNNs) are poised to make significant advances in the development of EMG-based hand prostheses for upper limb amputees [1]. However, the existing methods fail to fully satisfy practical requirements, as they can not simultaneously satisfy the requirements of accuracy and inference speed. Therefore, researchers have focused on investigating more accurate and efficient motor intention recognition methods from EMG signals [2]. Classification and continuous motion estimation are two common tasks in human-machine interaction (HMI). The classification can map actions into predefined discrete categories [3]. On the other hand, continuous motion estimation involves estimating continuous motion characteristics of actions from EMG signals. Compared to discrete classification, continuous motion estimation can more directly, naturally, and flexibly reflect motor intention. Continuous estimation can be performed with model-free methods [4], or musculoskeletal model-based methods [5]. Currently, deep learning is a promising approach for HMI and human-machine collaboration (HMC), and the pursuit of high accuracy and efficiency has become an important goal in the field. The RNN series models have been extensively used to estimate continuous hand kinematics from electromyographic signals. For example, a method for the estimation of hand pose from sEMG using an RNN structure is described in [6]. The estimation of continuous finger movements using a Long Short Term Memory (LSTM) neural network is proposed in [7]. A Long Exposure Convolutional Memory Network (LE-ConvMN) has also been proposed to predict the finger joint angles for multiple actions [8]. However, the RNN series models have a limitation in that they can not operate in parallel, resulting in lower model efficiency in practical usage. Another type of method used for EMG processing is the Convolution series models [9], such as the Temporal Convolutional Network (TCN) [10]. The convolution series models are very efficient, but their accuracy is low due to the instability of

the convolutional structure and the random characteristics of the sEMG signals. Recently, the Transformer series models have attracted great attention in the field of motion estimation from surface EMG [9]. In [2] the multi-feature Transformer is proposed to improve estimation accuracy, which is a milestone in continuous estimation from sEMG signals. In [11], the Bidirectional Encoder Representations for Transformers (BERT) is proposed to estimate continuous finger movements across subjects. As far as we know, almost all the presented work has been conducted with the aim of improving the accuracy of estimation, with less attention to efficiency. Model efficiency is a crucial research direction in deep learning [12]. In order to cope with the problem of large memory footprint as well as energy consumption, the Bioformer method [13], which is an embedding Transformer, was introduced to address the classification efficiency. A pruning VIT network has been previously proposed to identify high-density sEMG (HD-sEMG) signals reducing the required training samples in the training stage [14]. A Transformer-based gesture recognition algorithm for HD-sEMG has been proposed to achieve high accuracy but with high complexity [15]. For the task of continuous sEMG estimation, existing literature on Transformers [2] reports their accuracy and inference time on hardware devices such as ARMs (Advanced RISC Machines, Raspberry Pi 4B is a typical ARM), which does provide a detailed analysis of the model's deployability. In the latest study that performed cross-subjects testing, BERT [11][16] was evaluated as a large model on CPUs, but the higher arithmetic power of Intel's latest processor can not fully reflect its deployability on low-power embedding devices. LSTA-Conv is also a cross-subjects model and has been applied in EMG processing but without reporting the inference time on hardware [17]. In our preliminary investigation, we observed that the Transformer models are very accurate in estimating the continuous motion of finger joints, but very little work has been carried out on its efficiency. In this study, we focus on improving the efficiency of the Transformer method while concurrently preserving high accuracy in the continuous estimation of joint angles from sEMG signals.

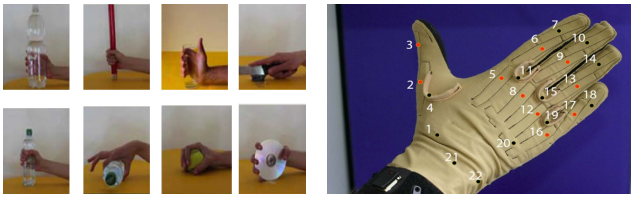
In order to meet the requirements of accuracy and efficiency concurrently, we propose two efficient Transformer models: sTransformer-EMSA and sTransformer-EMFN, that will be explained in details in the following. The proposed models were applied on the Ninapro DB2 dataset, and were compared with a variety of models commonly used in the field of continuous estimation. Not only we compared the accuracy of the models, but also their size, power dissipation, and the deployment time on a variety of edge devices. The Soft-dtw loss function [18] was adopted in the training stage to improve the accuracy of estimation. The experimental results demonstrated that the proposed models achieve high efficiency and accuracy. The paper's main contributions are as follows:

- sTransformer-EMSA: The EMSA with downsampling mechanism is proposed in Transformer to improve the efficiency of continuous estimation.
- sTransformer-EMFN: The fusion strategy of EMSA with down-sampling and Feedforward Neural Network (FNN) with pruning in Transformer is proposed, aiming at improving the efficiency of the model.
- The Soft-dtw loss function is adopted to improve the accuracy of the model in the training stage.

II. MATERIAL AND METHODS

A. Dataset and Feature Extraction

Ninapro [19] is an open-source dataset of EMG signals. Ninapro employs the Ottobock 13E200-50 and Delsys Trigno Wireless EMG systems (using 12 wireless sEMG electrodes) and the CyberGlove II data gloves (using a 22-sensor to capture the joint angles). These data acquisition tools have a sampling rate of 2kHz. The Ninapro DB2 includes data from 40 healthy subjects (28 men, 34 right-handed). In our experiments, data from 38 subjects were used for presenting the results (data from two subjects were excluded because judged as corrupted) .



(a) (b)

Fig. 1. Eight grasping movements (a) with items such as water glasses, discs, and tennis balls are used in continuous motion estimation and CyberGlove II data-glove (b). The red dot represents the degrees of freedom of the ten joints [8]

We selected the eight common grasping movements shown in Fig. 1 (a), and the 10 finger joints as shown in Fig. 1 (b). In the published papers [8][11], six typical continuous grasping actions are selected for estimation, we follow the same way, that is, we selected the same eight grasping actions as in [8][11] to estimate. Inspired by [8], we used long-exposure to elongate the data extraction range and enhance the generalization of the data. Root means square (RMS) [20] was used to extract data features. We set the duration of the time window for RMS calculation to 100 ms and the step length to 0.5 ms.

In the Ninapro dataset, each grasping action is repeated six times, and the corresponding sEMG signals and joint angle information are recorded for each repetition (trial). Each action was further divided into a training dataset, consisting of four trials, and a testing dataset, comprising two trials.

B. Data normalization

The μ-law normalization [21] is often used in audio processing, mainly to logarithmically amplify low-frequency audio signals and to improve the generalization of the models to low-frequency features. Previous work has demonstrated that the μ-law normalization works well for sEMG signals [11]. The μ-law normalization is performed as follows:

$$F(x_t^i) = \text{sign}(x_t^i)\frac{ln(1+\hat{\mu}|x_t^i|)}{ln(1+\hat{\mu})} \quad (1)$$

where x_t^i represents the sEMG data for i.th channel, and $\hat{\mu}$ represents the hyperparameters which we need to set. In addition to μ-law normalization, we also used z-zero (zero-mean) normalization to compare with [22]. The temporal characteristics of sEMG signals vary greatly between samples, and some abnormal values may affect the training of the model. z-zero normalization can normalize the standard deviation of the feature values so that RMS-extracted feature data conforms to a normal distribution and the training efficiency of the model improves. The z-zero normalization is given by:

$$Z(x_t^i) = \frac{x_t^i - \mu^i}{\sigma^i} \tag{2}$$

where x_t^i represents the sEMG data for i.th channel, μ^i is the mean value of sEMG data for each channel, and σ^i is the standard deviation of sEMG data for each channel.

C. Performance parameters

The Pearson correlation coefficient (PCC) was used to measure the correlation between the actual and the predicted joint angles. In this paper, we used PCC to evaluate the models. It was computed as follows:

$$CC = \frac{\sum_{t=1}^{N}\left(\theta_{pred} - \overline{\theta_{pred}}\right)\left(\theta_{real} - \overline{\theta_{real}}\right)}{\sqrt{\sum_{t=1}^{N}\left(\theta_{pred} - \overline{\theta_{pred}}\right)^2}\sqrt{\sum_{t=1}^{N}\left(\theta_{real} - \overline{\theta_{real}}\right)^2}} \tag{3}$$

where θ_{pred}, $\overline{\theta_{pred}}$, θ_{real}, $\overline{\theta_{real}}$ denote the model-predicted joint angle value, the average value of the model-predicted joint angle, the real joint angle value, and the average value of the real joint angle.t represents the observation time, and θ_{pred}, $\overline{\theta_{pred}}$, θ_{real}, $\overline{\theta_{real}}$ are the functions of t.

The root means square error (RMSE) [23] can be used to measure the error between the actual joint angles and the joint angles predicted by the deep learning model. RMSE can be used as one of the criteria for judging the merit of the model. RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(\theta_{pred} - \theta_{real}\right)^2} \tag{4}$$

Since the range of activity of each finger is subject-specific, it is not possible to use RMSE to evaluate the merit of the algorithm uniformly, so we used the normalized value NRMSE as an evaluation index:

$$NRMSE = \frac{\mathrm{RMSE}}{\theta_{max} - \theta_{min}} \tag{5}$$

where θ_{max} and θ_{min} represent the maximum and minimum values of actual joint angles respectively.

The computational power is also a very important evaluation metric in deep learning, and we used the real ARM device to evaluate how much computational power the models consumed. We used the ratio of inference time to model power consumption as an evaluation metric, called ARM Model Efficiency (AME):

$$\mathrm{AME} = \frac{Infer\_Time}{Power} \tag{6}$$

where *Infer_Time* is the inference time for model deployment, and *Power* is the power consumption of the offline operating model for ARM devices.

III. THE MODEL FRAMEWORK

The details of the structure of the proposed model are presented in this section. The proposed model for continuous estimation from sEMG signals is shown in Fig. 2, including feature extraction (RMS), model prediction, and smooth layer. The structure of the model contains the following main modules: the positional encodings, the encoder-decoder, EMSA, and the pruning FNN module, as shown in Fig. 3. Compared with the standard Transformer, the proposed model, which is based on EMSA and the pruning mechanism, achieves high accuracy and efficiency.

A. Positional Encoding

In 2017, Transformer was proposed in the natural language processing (NLP) field with its Muti-Head self-attention structure [24]. Recently, Transformer has been used in the field of myocontrol to perform gesture classification and continuous estimation tasks. In our model, the input 12-channel sEMG signal is first positionally encoded. We used the same positional encoding method as in [24]:

$$P_{(pos,2i)} = \sin\left(\frac{pos}{1000^{2i/d}}\right) \tag{7}$$

$$P_{(pos,2i+1)} = \cos\left(\frac{pos}{1000^{2i/d}}\right) \tag{8}$$

where d represents the dimension of the vector, pos and i are the position and dimension of the input sEMG signals, respectively. Each dimension of the position encoded sEMG signal corresponds to a sinusoid, with wavelengths forming a geometric series of 2π to $1000{\cdot}2\pi$.

The sEMG after RMS feature extraction $X_{input} = [x_0, x_1, x_2, ..., x_m]$ is encoded at the input layer, where X_{input} represents the 12 channels$*$100ms RMS features of sEMG signals. Here, m is 199. We linearly expand X_{input} according to the number of hidden layers, which helps to extract the spatial information of the sEMG signals. The expanded data is designated as X_{emb}:

$$X_{emb} = \mathrm{Linear}\left(X_{input}\right) \tag{9}$$

We extract temporal information from X_{emb} by projection embedding:

$$X_{PE} = \left[x_1^{emb}, x_2^{emb}, ..., x_m^{emb}\right] + T_{pe} \tag{10}$$

where x_m^{emb} represents projection embedding of X_{emb} in linear layer, T_{pe} is the time vector after encoding the position in equations (7) and (8), which gives the model the ability to capture the temporal information from sEMG features, X_{PE} is incorporation of X_{emb} and location information.

B. Efficient Mult-Head Self-Attention (EMSA)

The EMSA module [25] can improve the efficiency of standard Transformer. The EMSA mechanism consists of a Self-attention mechanism, downsampling of the convolution mechanism, and a layer of normalization. Self-attention can enhance the ability to characterise EMG signal features and

This article has been accepted for publication in IEEE Transactions on Neural Systems and Rehabilitation Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2024.3514938

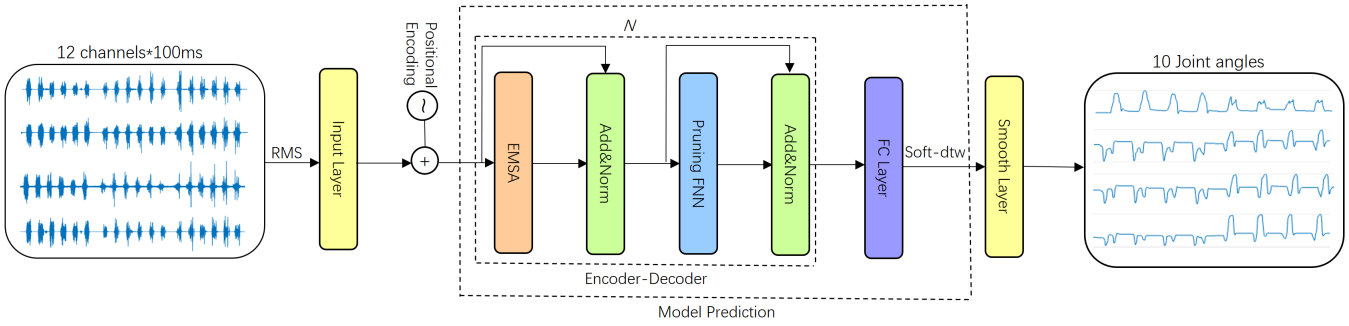
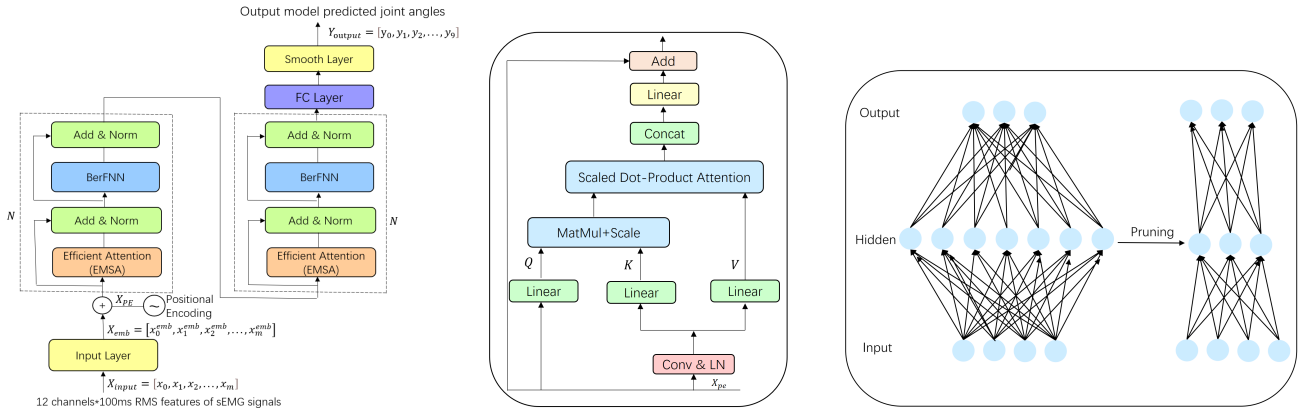


Fig. 2. Model structure based on the transformer approach for continuous kinematic estimation of finger joints. The raw sEMG signal is input to the model through feature extraction, position coding, efficient multiple self-attention mechanism, and pruning FNN layer, the smoothing layer, and finally the prediction of the joint angle. RMS refers to the way in which features are extracted. N represents the number of layers in the model.



(a) Encoder-Decoder (b) EMSA (c) Pruning FNN

Fig. 3. The detailed structure of transformer module: (a) Encoder-Decoder module (b) EMSA module (Downsampling of K,V key values using 2D convolution), (c) Pruning FNN module (Pruning of hidden layer neurons in the FNN layer)

to understand the relationship of sEMG signals sequence between different channels. Convolutional operation is executed to extract the spatial information of sEMG signals. Layer normalization which can be addressed by the proposed model degradation is applied to the module to make the output distribution of the residual connections layer.

The Self-attention mechanisms included queries Q_h, keys K_h, and values V_h. The transpose of K_h and Q_h was then subjected to a dot product operation, followed by a softmax operation to update the weights, and then multiplied by dot product with the matrix V_h to get the result of self-attention:

$$\text{Self-attention}\left(Q_h, K_h, V_h\right) = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_{kk}}}\right) V_h \quad (11)$$

where the d_{kk} is the dimension of the keys. Multi-headed attention allows the model to use different attention heads to focus on the relations of the sEMG of inputs. The convolutional operation is adopted in EMSA to downsample K_h, V_h.

The downsampling times of the convolution are derived from the size of the convolution kernel, stride, and padding. We regulated the parameters of convolution to balance the computational accuracy and efficiency of the model. For simplicity, we set the downsampling times to a fixed value. The procedure of downsampling can be described as follows:

$$X_c = \text{Conv}(X_{pe}) \quad (12)$$

$$X_l = \text{LayerNorm}(X_c) \quad (13)$$

where $X_{pe} \in \mathbb{R}^{c \times h \times w}$ is reshaped by the $X_{PE} \in \mathbb{R}^{c \times l}$ which is the projection embedding from the sEMG signal. X_c is the output of the convolutional operation, X_l is the output of the layer normalization, c and l are the number and length of the channel, respectively, h and w are the height and width of the sEMG signals features, respectively.

In addition, the model uses the classical encoder-decoder architecture of Transformer, using the output of the encoder as the input to the memory decoder for fitting the joint angles, which enhances the fitting ability of the model.

C. *Adaptive Pruning FNN*

The internal structure of EMSA is mainly matrix multiplication, thus it performs linear transformations. The learning ability of linear transformations is not as strong as that of nonlinear transformations. EMSA can learn a new representation of the sEMG signals, while the representation might not be so strong. The FNN layer was used to enhance this representation after EMSA. The number of neurons in the FNN layer has a huge impact on the performance of the model and can be regulated according to demands.

To alleviate the pressure and improve the performance of the model [26] [27], we tried to set the number of neurons in FNN, which is the layer with adaptive pruning. The binary gate

g_t, which controls the granularity of hidden units, was sampled from Bernoulli, a distribution with a learnable parameter α_p that can be automatically optimized by the gradient. The g_t is obtained as:

$$g_t \sim \mathrm{Ber}\left(\mathrm{Sigmoid}\left(\alpha_{\mathrm{p}}\right)\right) \tag{14}$$

Sigmoid(α_p) is the activation function to calculate the selecting possibility of the i-th neuron. The purring FNN can be expressed as:

$$FNN\left(X\right) = \sum_{i \in c_{in}} g_t \mathrm{Gelu}(XW^{fc1})W^{fc2} + b \tag{15}$$

where $W^{fc1} \in \mathbb{R}^{c_{in} \times c_{in'}}$ and $W^{fc2} \in \mathbb{R}^{c_{in'} \times c_{in}}$ are a full-connected layer that can be learnable, c_{in} denotes the predefined dimensions, and $c_{in'}$ denotes the FNN hidden dimensions, which can be learned through pre-training. The adaptive pruning mechanism ensures the efficiency of the pruning.

D. Smoothing Layer

The Transformer series model was originally designed for NLP, and when using it in predicting finger joint angles, some fluctuations are introduced. We set up a smoothing module after training module to reduce fluctuation. The smoothing module aims to mitigate the oscillations in the model by averaging the samples within the sliding window. The smoothing layer is given as:

$$\mathrm{AvgSmooth}(X) = \left[\sum_{i=1}^{w} x_i/w, \sum_{i=2}^{w} x_i/w, ..., \sum_{i=n}^{w} x_i/w\right] \tag{16}$$

where w is the size of the sliding window, and $X = [x_1, x_2, ..., x_i]$ is the predicted value of the finger of the hand joint angles by the model.

E. Distance-based Loss Function: Soft-dtw

In previous models of continuous estimation, we have often used the mean square error (MSE), a Euclidean distance, to measure the difference between the real and the estimated angles. MSE is a loss function based on the Euclidean distance. However, the Euclidean distance-based loss function is vulnerable to fluctuations, such as noise and some other uncertainties.

Dynamic Time Warping (DTW) [28] is a method of calculating the similarity between two sequences, which can be regularised according to the time dimension and waits until the predicted results are better matched. The DTW is a nondifferentiable process. Cuturi & Blondel [18] designed a Soft minimum instead of the DTW minimum to get a differentiable Soft-dtw.

For the sequences of real and estimated joint angles, the output of the model is $\overline{Y} = (\overline{y_1}, \overline{y_2}, ..., \overline{y_n}) \in \mathbb{R}^{p*n}$, where p and n are the length and width of the sequences, respectively. The actual sequence is $Y = (y_1, y_2, ..., y_n) \in \mathbb{R}^{p*n}$. Let's define the set $R = [r_{i,j}], R \in \mathbb{R}^{n*n}$ as the cost matrix, and $r_{i,j}$ as the component of the cost matrix. By using the algorithm of dynamic programming, the Soft-dtw can be defined as:

$$r_i^j = \delta(i,j) + min^{\gamma}\left\{r_{i,j-1}, r_{i-1,j}, r_{i-1,j-1}\right\} \tag{17}$$

$$min^{\gamma} y_1, ..., y_n = \begin{cases} min_{i \le n} y_i, \gamma = 0 \\ -\gamma log \sum_{i=1}^{n} e^{\frac{-y_i}{\gamma}} \gamma > 0 \end{cases} \tag{18}$$

where δ is the distance function. γ is a smoothing parameter.

This is the first time that a loss function from a time series has been applied to the continuous estimation of finger joints from sEMG signals.

IV. EXPERIMENTS AND RESULTS

We conducted experiments using different variants of the proposed models and compared them with the commonly used models in the field of hand kinematic estimation from sEMG signals. The proposed model uses PyTorch 1.8.0 [29] and was trained on an NVIDIA GeForce RTX 3090 GPU. The time window for all models was set to 100 ms, and the step size was set to 50 ms. The Adam optimizer was used for all models, with a learning rate of 0.0001 and a batch size of 32 for training. Two types of normalization u-law normalization and z-zero normalization are conducted in experiments.

In the experiments, CC, RMSE, and NRMSE are indicators that measure the performance of the model in predicting the joint angles. For each movement, there are 6 trials, we select 4 trials for training and 2 trials for testing, we perform cross-validation, C_6^4 trails were selected for training, and the left C_6^2 trails for testing. Our deep learning model predicts eight kinds of consecutive grasping action then averages the results over 38 subjects. The amount of model parameters, which represents the size of the model, inference time, and AME, are metrics for evaluating the efficiency of the model on the hardware device. Specifically, we measured the power consumption on the ARM device and combined AME to evaluate the efficiency of the model on the power aspect. The amount of the parameters in the Transformer series model was saved and used as a reference for model size. For these results, the Friedman test and the Wilcoxon signed-rank test were used to assess the significance of the proposed model.

We trained the convolution series, RNN series, and Transformer series models for eight movements on 38 subjects to evaluate the performance of different models, Table I shows the results normalized to μ-law, the hyperparameters $\hat{\mu}$ was set to 2^{20} in our study for consistency with the previous study [11], and Table II shows the results normalized to z-zero, with the model prefix 's' indicating the use of a smoothing layer, the suffix '-EMSA' after the model indicates the use of an efficient multiple self-attention mechanism, "-win" denotes the utilization of a sliding window algorithm in the multi-headed attention mechanism, '-EMFN' indicates the use of an efficient attention mechanism and pruning of the FNN layer, and '* ' denotes a model trained using the Soft-dtw loss function, and 'LE' denotes a model using the long-exposure mechanism.

TABLE I
AVERAGE PERFORMANCE OF DIFFERENT SERIES MODELS ON CONTINUOUS ESTIMATION OF HAND KINEMATICS ON NINAPRO DB2 WITH μ-LAW

Model	CC	RMSE	NRMSE	Epoch Time(s)
Convolution Series Models				
TCN	0.6805	14.11	0.1511	2.53
LS-TCN	**0.7040**	**13.43**	**0.1437**	2.56
RNN Series Models				
LSTM	0.6221	15.019	0.1597	6.20
GRU	0.6217	17.54	0.1863	8.10
LE-LSTM	0.6871	16.40	0.1737	**46.00**
LE-ConvMN	**0.8338**	**10.48**	**0.1138**	27.22
Transformer Series Models				
sBERT	0.7966	11.67	0.1261	5.07
sBERT-EMSA	0.7993	**11.43**	**0.1236**	5.04
sTransformer	0.7820	12.21	0.1319	4.97
sTransformer-win	0.8003	11.85	0.1278	6.80
sTransformer-EMSA	0.7920	11.76	0.1267	5.13
sTransformer-EMSA*	0.7930	12.24	0.1321	5.65
sTransformer-EMFN	**0.8030**	12.06	0.1302	5.05

1. Table I presents the experimental results obtained using the μ-law normalization function.
2. 's' denotes a model using a smoothing layer, 'win' denotes a sliding window applied in the attention mechanism, 'EMSA' denotes the efficient multiple self-attention mechanism, and 'EMFN' denotes the fusion of EMSA with pruning FNN layers. The marker $*$ represents the Soft-dtw loss function is adopted for training the model.

TABLE II
AVERAGE PERFORMANCE OF DIFFERENT SERIES MODELS ON CONTINUOUS ESTIMATION OF HAND KINEMATICS ON NINAPRO DB2 WITH Z-ZERO

Model	CC	RMSE	NRMSE	Epochs Time(s)
Convolution Series Models				
TCN	0.7713	12.24	0.1390	2.50
LS-TCN	**0.7844**	**12.02**	**0.1314**	2.56
RNN Series Models				
GRU	0.7421	12.52	0.1337	6.02
LSTM	0.7437	12.39	0.1326	7.80
LE-LSTM	0.8030	11.58	0.1244	**46.96**
LE-ConvMN	**0.8556**	**9.580**	**0.1034**	28.03
Transformer Series Models				
sBERT	0.7916	13.51	0.1452	5.05
sBERT-EMSA	0.7921	12.97	0.1390	5.03
sTransformer	0.8206	10.68	0.1151	4.98
sTransformer-win	0.8171	10.86	0.1167	6.73
sTransformer-EMSA	0.8213	10.71	0.1152	5.10
sTransformer-EMSA*	**0.8234**	**10.66**	**0.1145**	5.60
sTransformer-EMFN	0.8177	10.77	0.1158	5.01

1. Table II illustrates the results obtained using z-zero normalization function.
2. 's' denotes a model using a smoothing layer, 'win' denotes a sliding window applied in the attention mechanism, 'EMSA' denotes the efficient multiple self-attention mechanism, and 'EMFN' denotes the fusion of EMSA with pruning FNN layers. The marker $*$ represents the Soft-dtw loss function is adopted for training the model.

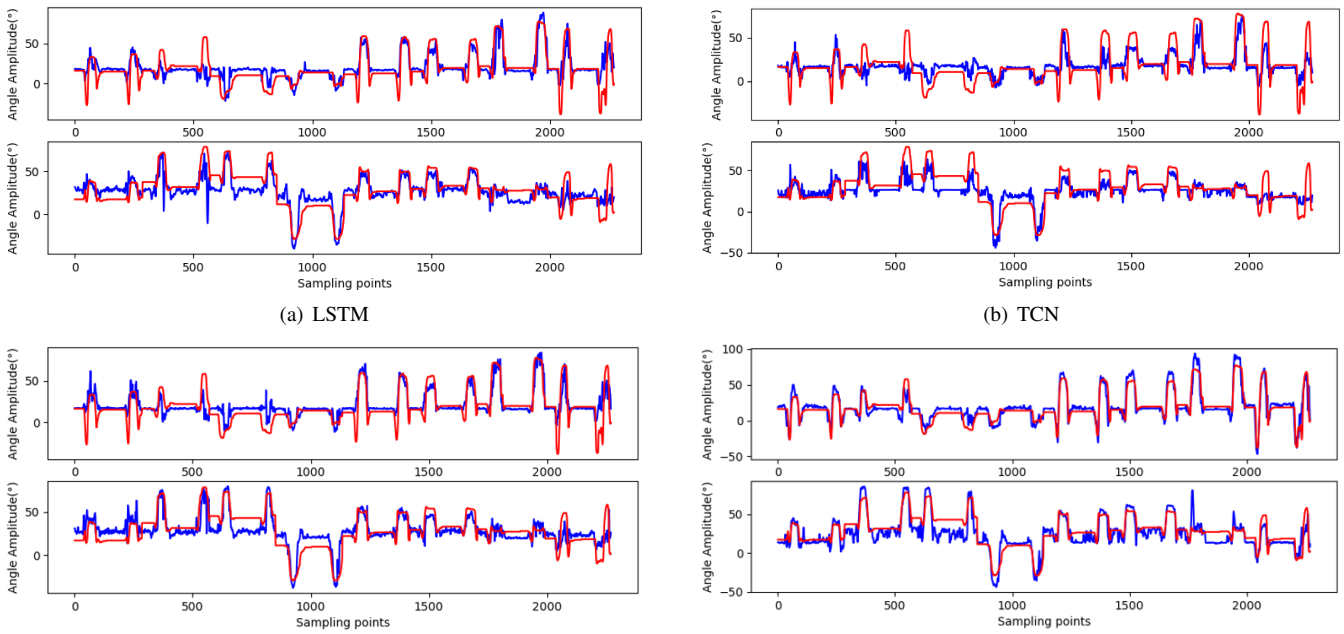
Comparing Table I and Table II, the z-zero normalization was better than the u-law normalization for accuracy. We found that the model sTransformer-EMSA* in Table II, which is trained with the Soft-dwt loss function (unpruned), achieved the highest performance (CC = 0.8234 ± 0.04; RMSE = 10.66 ± 1.45; NRMSE = 0.1145 ± 0.01). The second best model is sTransformer-EMFN (CC = 0.8177 ± 0.04, RMSE = 10.77 ± 1.48, NRMSE = 0.1158 ± 0.01), which was pruned. The accuracy of LE-ConvMN was higher than sTransformer-EMSA* and sTransformer-EMFN, but the training time of LE-ConvMN was also longer. From Table II, we found that the average CC, RMSE and NRMSE of the proposed methods (sTransformer-EMSA, sTransformer-EMFN) of was better . than TCN (CC=0.7713 ± 0.05, $p < 0.001$; 12.24 ± 1.62, $p < 0.001$; 0.1390 ± 0.01, $p < 0.001$), better than LS-TCN (CC = 0.7844 ± 0.04, $p < 0.001$; RMSE = 12.24 ± 1.48, $p < 0.001$; NRMSE = 0.1314 ± 0.01, $p < 0.001$), better than sBERT (CC = 0.7916 ± 0.05, $p = 0.003$; RMSE = 13.51 ± 1.80, $p < 0.001$; NRMSE = 0.1452 ± 0.02, $p < 0.001$),better than LSTM (CC = 0.7437 ± 0.04, $p < 0.001$; RMSE = 12.39 ± 1.80, $p < 0.001$; NRMSE = 0.1326 ± 0.01, $p < 0.001$), better than LE-LSTM (CC = 0.8030 ± 0.05, $p = 0.076$; RMSE = 11.58 ± 1.81, $p = 0.027$; NRMSE = 0.1244 ± 0.02, $p = 0.003$), but worse than LE-Conv-MN (0.8557 ± 0.03; RMSE = 9.580 ± 1.30; NRMSE = 0.1034 ± 0.01). LE-convMN is currently the approach with the best single individual accuracy.

Among 38 subjects, subject S5 performed the best. Fig.4 shows the results with two representative angles in the classical network and our proposed Transformer-EMFN.

In terms of training time, the μ-law normalization method took longer to achieve convergence compared to z-zero. In our model, z-zero converged after approximately 150 epochs, while μ-law took approximately 250 epochs to converge. We found that the long-exposure series of the models was the slowest, with about 1000 epochs. The LE-Conv_MN requires recording the information from the previous training of the model and uses it in the next training, which prolongs the training time. On the other hand, TCN, which is a typical convolution series model, was the fastest due to the nature of convolution, which allows for parallel computing. The local connection within the TCN further speds up the training process. The Transformer series model was larger, and the training speed between different Transformers was negligible. The feature of parallel computing can accelerate the training of Transformer models, while it can not be utilized in practice.

Table III shows the inference time of different models on different devices. We deployed the models on an Inter CPU I5-7300HQ, an Apple M1 chip, and a Raspberry Pi 4B device to measure the inference time in real applications. The model inputs are the same sEMG signals, the concrete input format is 12 channels*100ms with a 2000Hz sampling rate, that is, a 12*200 data matrix. The past 12 channels*100ms is necessary for RMS processing, it is because the step size in RMS in our simulation is 1 sampling point per time. The output of all models is a joint angles vector in 10*1, corresponding to the 12*200 data matrix. There is a 150ms delay between the predicted joint angles and the real joint angles because the step size of the sliding time window is 150ms. We record the computing time from inputting to outputting as inference time on all devices. Among them, the I5-7300HQ (maximum power consumption 45w) and Apple M1 (maximum power consumption 39w) are computer CPUs

This article has been accepted for publication in IEEE Transactions on Neural Systems and Rehabilitation Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2024.3514938



(a) LSTM (b) TCN

(c) BERT (d) Transformer-EMFN

Fig. 4. The graph shows the results of our model based on the z-zero standard discourse approach on a single individual. The figure shows two predicted joints (Middle finger proximal interphalangeal joint and Middle finger metacarpophalangeal joint), and three models for comparison, where the blue curve is the result predicted by the model and the red curve is the actual result.

known for their high computing ability but also high chip power consumption,making them difficult to be deployed in wearable devices. On the other hand, Raspberry Pi 4B is an ARM-based device with a maximum power consumption of only 6.25W and requires only a 5V power supply. We can assess whether a model is suitable to be deployed in wearable devices based on the comparison of inference time and power consumption. The experimental results in Table III show that TCN (2.11ms, 1.08ms, 8.06ms, $p < 0.001$) is the fastest among the convolution series models; LE-ConvMN (13.35ms, 8.54ms, 100.04ms, $p < 0.001$) is the fastest among the RNN series models; our model (13.28ms, 11.30ms, 65.99ms, $p < 0.001$) is the fastest among the Transformer series models. TCN is the fastest model on multiple devices, but its accuracy is the lowest, which limits its application in practice. The accuracy of our model is as similar as that of LE-ConvMN, while the inference speed of our model is much faster than that of LE-ConvMN. The Transformer series models can be trained in parallel, that is an advantage that LE-ConvMN does not have. We conclude that our model is the most promising in wearable applications.

The power consumption of Inter CPU and Apple's M1 chip are too high for wearable applications, so we focused on testing the Transformer series models on the Raspberry Pi 4B. Table IV shows the model efficiency of different Transformer series models on Raspberry Pi 4B with z-zero normalization. We can find that BERT has the biggest model size, the longest inference time, and the highest AME, it is mainly due to BERT adopts learnable vector instead of positional encoding, the learnable vector contains too many linear layers. In contrast, our downsampling of K and V and the pruning of the unimportant hidden neurons in the FNN layer reduces the computational density and the amount of parameters, leading to lower power consumption, the lowest AME, smallest model size and the fastest inference speed among all the Transformer series models. The proposed model is slightly less accurate than the baseline model sTransformer,but the efficiency has been greatly improved.

We conducted experiments with DB7 and SEEDs to

TABLE III
DIFFERENT INFERENCE TIME OF DIFFERENT SERIES MODELS ON CONTINUOUS ESTIMATION OF HAND KINEMATICS ON NINAPRO DB2 WITH Z-ZERO NORMALIZATION

Model	Intel CPU	Apple M1	Raspberry Pi 4B
Convolution Series Models (ms)			
TCN	**2.11**	**1.08**	**8.06**
LS-TCN	4.4	2.37	24.42
RNN Series Models (ms)			
GRU	25.75	9.12	117.13
LSTM	25.91	10.49	137.96
LE-LSTM	27.21	10.72	138.67
LE-ConvMN	**13.35**	**8.54**	**100.04**
Transformer Series Models (ms)			
sBERT	39.34	36.20	336.99
sBERT-EMSA	32.51	28.25	301.89
sTransformer	30.22	16.31	116.43
sTransformer-swin	54.19	28.22	216.80
sTransformer-EMSA	22.61	12.25	102.55
sTransformer-EMFN	**13.28**	**11.30**	**65.99**

1. The inference time of different models on different hardware devices were tested, and results are the average of several subjects.
2. The CPUs used in the table are the Intel microprocessor I5-7300HQ, Apple's M1, and the Raspberry Pi 4B. Intel and Apple are mainly computer laptop CPU devices and Raspberry Pi 4B is a microcomputing processor device.

TABLE IV
MODEL EFFICIENCY OF DIFFERENT TRANSFORMER SERIES MODELS ON RASPBERRY PI 4B WITH Z-ZERO NORMALIZATION

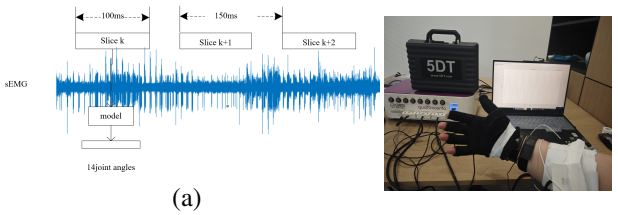
Model	Params(MB)	Infer(ms)	Power(W)	AME
sBERT	59.63	336.99	2.20	153.18
sBERT-EMSA	59.5	301.89	**2.10**	143.76
sTransformer	3.43	116.43	3.10	37.56
sTransformer-swin	3.53	216.80	2.40	90.33
sTransformer-EMSA	3.43	102.55	3.00	34.18
sTransformer-EMFN	**2.95**	**65.99**	2.80	**23.57**

1. The batch_size is set to 1 and the window length of input is set to 200 points.
2. The device used to measure the inference time is the Raspberry Pi 4B.

explore the effects of noise and electrodes shifting on our model. In DB7, when Gaussian noise with SNR 10 dB is added, the accuracy of Transformer-EMFN is decreased by only 0.5 %. This preliminary proves the ability of our model to resist noise. In SEEDS[30] datasets , we conduct the electrodes shifting simulation. We used the odd rows of electrodes for training and the even rows of electrodes for testing to simulate the shifting of the electrodes. After Electrodes's shifting, the accuracy of Transformer-EMFN (CC = 0.7610, RMSE = 12.25, NRMSE = 0.1324) is decreased by 5.12% comparing with no electrodes' shifting. Finally, we came to record the model inferring time on DB7 with ARM PI 4B. The inferring time of Transformer-EMSA and Transformer-EMFN is 105.78ms and 67.35ms, respectively, which is promising for real-time applications.

We also executed online test: High-density sEMG was acquired from forearm muscles utilizing three grids (ELSCH064NM3, 8 × 8 channels, OT Bioelettronica, Italy) which was connected to a multichannel amplifier (QUATTRO-CENTO, OT Bioelettronica, Italy). The sEMG signals (192 channels) were recorded in monopolar derivation with a gain of 500 and sampled at 2048 Hz. Following recording, the signals underwent bandpass filtering with cut-off frequencies set between 10 and 500 Hz, and A/D conversion on 12 bits. Simultaneously, finger kinematics parameters were captured using a 5DT Data Glove 14 Ultra (5DT Inc. USA). Synchronization between the sEMG recorder and data glove was achieved through a MATLAB program (Figure 5(b)). The data glove measures the angles of 14 finger joints at a sampling rate of 16 Hz. Six movements (3-digit pinch, cylinder grasp, disc grasp, key grip, pinch and grasp, fist) were chosen, each movement executed for 30s, and 20s is used as the training set for training the models, and the remaining are used as testing set to evaluate models. The model was combined in the online experimental system. A sliding time window with a step size of 150ms and a window length of 100ms is used to estimate the joint angles from sEMG signals in Figure 5 (a), before estimation, the RMS is calculated within this sliding window, the step size of calculating RMS is 0.5ms. Four able-bodied subjects (all male, aged 32 ± 5 years) participated in the experiment. The results are shown in Table V.

From TABLE V our model performed well in online test, (PCC=0.8028±0.08, RMSE=12.02±0.45, NRMSE=0.1298±0.22), the inference time (ms) for our model on sEMG with 192 channels (Intel CPU=14.35±0.25, Apple M1=10.20±1.47, Raspberry Pi 4B=105.36±5.58) is relevantly short, and the model efficiency (Params=13.2MB, Power=2.30±0.27W, AME=45.81±4.40) can satisfy the demands of real applications.



(a) (b)

Fig. 5. The online testing procedure. (a) Sliding time windows. (b) Online experiment setup

TABLE V
AVERAGE ACCURACY, INFERENCE TIME ON INTEL CPU, APPLE M1 AND RASPBERRY PI 4B AND MODEL EFFICIENCY FOR OUR MODEL ON FOUR SUBJECTS IN ONLINE EXPERIMENT

Accurac	CC	RMSE	NRMSE
	0.8028	12.02	0.1298
Inference Time (ms)	Intel CPU	Apple M1	Raspberry Pi 4B
	14.35	10.20	105.36
Efficiency	Params	Power	AME
	13.2MB	2.30W	45.81

In the online experiment, we set the step size at 150 ms, the reason is the inference time of our model is 14.35 ms and the time of data filtering, calculation of RMS, and other time costs are about 120 ms. In the online experiment, less inference time can lead to a smaller step size of the time window, which can get more estimated angles during the same time. More estimated angles can improve the smoothness of human-computer interaction and reduce the possibility of losing critical information. Though the inference time of LE-ConvMN is only 100.04ms on sEMG with 12 channels, the inference time on HD-sEMG (192 channels) is 653.30ms, which is too long to be accepted in HIM. Generally, the models with faster inference time always suffer from lower accuracy, while our model combines fast inference with high accuracy, which makes it a good choice for real-time applications.

In conclusion, among the Transformer series models, the proposed model with smoothing layers, EMSA, and pruning FNN layers, yielded the best results. Among the convolution series models, TCN had the fastest inference speed but suffered from low accuracy. LS-TCN had improved the accuracy, but it still fell short of practical demands. Among RNN series models, recurrent neural networks achieved good accuracy but suffered from an inability to train in parallel, which leading to longer training time and inference time. The classical Transformer model can be trained in parallel, and there is still room for improvement in the number of parameters and inference time. Our model outperformed TCN, LS-TCN, and LE-LSTM in all metrics, including PCC, NRMSE, inference time, and power consumption. Although LE-Conv-MN had the accuracy advantage, our model outperformed it in terms

of training time, inference time, and power consumption. In summary, our model can be trained in parallel, with shorter training and inference time and lower power consumption, making it have strong practical value.

V. Discussions

In the current study, we proposed the sTransformer-EMSA and sTransformer-EMFN methods to continuously estimate finger joints kinematics from sEMG signals. We compared the performance of Convolution series, RNN series and Transformer series models, with metrics such as CC, RMSE, NRMSE, and training time per epoch. We also calculated the inference time for a window of input during the testing phase, the model size, power consumption, and AME. Our results demonstrated that the Transformer series model outperformed classical models. Ablation experiments further showed that the proposed approach outperformed the classical Transformer model. Our method achieved a balance between efficiency and accuracy, with shorter inference time, smaller model size, lower power consumption, and smaller loss of accuracy compared to the classical Transformer model, making it more promising for practical applications.

We used efficient attention to decrease the spatial complexity of the model, resulting in reduced inference time and improved training efficiency. For accuracy, our approach is comparable to those of SOTA methods, but with shorter training time and inferring time, lower power consumption, and smaller model size.

There are several limitations to our work. First of all, although we included all individuals in Ninapro Dataset DB2, our model was only tested on the single subject and not on cross-subjects [11]. The transfer learning on cross-subjects is a researching trend, and it will be our future work [31]. In addition, in this paper, the models based on Transformer consume too much arithmetic power in the training stage. In the future, we will research the possibility of using integral operations instead of the floating-point operations in Transformer to improve the accuracy and efficiency further.

In future research, our focus will not only be on model compression but also on model quantization [32] . Currently, model quantization for specific hardware devices has become a mainstream in industrial applications, which significantly impacts the practical application of models. Quantization involves converting the floating-point operation into integral operation, resulting in a substantial reduction in the amount of model parameters and improved inference time. However, quantization may reduce the accuracy of the model. Therefore, how to quantize the model with minimum loss of accuracy is an important research task. Multimodality has also begun to emerge in various fields.

VI. Conclusions

The continuous estimation of movement from sEMG signals has been a relevant topic in myoelectric control and more generally in human-machine interfacing research. Previous work almost exclusively focused on decoding accuracy. In this paper, we specifically focused on developing models that simultaneously provided high accuracy and efficiency. We proposed the sTransformer-EMSA and sTransformer-EMFN models, which utilize efficient multiple self-attention to replace the traditional self-attention mechanism, and pruning FNN strategy to improve algorithmic efficiency. The models achieved an effective myoelectric homogeneous estimation of joint angles during grasping movements. By optimizing attention mechanisms and pre-processing, the models could extract the temporal and spatial features from sEMG signals, and the smooth layer facilitated superior accuracy of joint angles. Our results from the widely used Ninapro dataset demonstrated that the accuracy of sTransformer-EMSA and sTransformer-EMFN is as high as those of SOTA methods, and the efficiency is significantly better than previous methods. Moving forward, we aim to enhance the attention mechanism to bolster the efficiency and accuracy of the models and explore the possibility of deploying edge computing to further improve the comprehensive performance of the models.

References

[1] R. Akhundov, D. J. Saxby, S. Edwards, S. Snodgrass, P. Clausen, and L. E. Diamond, "Development of a deep neural network for automated electromyographic pattern classification," *Journal of Experimental Biology*, vol. 222, no. 5, p. jeb198101, 2019.

[2] W. Guo, N. Jiang, D. Farina, J. Su, Z. Wang, C. Lin, and H. Xiong, "Multi-attention feature fusion network for accurate estimation of finger kinematics from surface electromyographic signals," *IEEE Transactions on Human-Machine Systems*, vol. 53, no. 3, pp. 512–519, 2023.

[3] A. Stango, F. Negro, and D. Farina, "Spatial correlation of high density emg signals provides features robust to electrode number and shift in pattern recognition for myocontrol," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 2, pp. 189–198, 2014.

[4] M. Xiloyannis, C. Gavriel, A. A. Thomik, and A. A. Faisal, "Gaussian process autoregression for simultaneous proportional multi-modal prosthetic control with natural hand kinematics," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1785–1801, 2017.

[5] V. Sholukha, B. Bonnechere, P. Salvia, F. Moiseev, M. Rooze, and S. V. S. Jan, "Model-based approach for human kinematics reconstruction from markerless and marker-based motion analysis systems," *Journal of biomechanics*, vol. 46, no. 14, pp. 2363–2371, 2013.

[6] F. Quivira, T. Koike-Akino, Y. Wang, and D. Erdogmus, "Translating semg signals to continuous hand poses using recurrent neural networks," in *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pp. 166–169, IEEE, 2018.

[7] C. Wang, W. Guo, H. Zhang, L. Guo, C. Huang, and C. Lin, "semg-based continuous estimation of grasp movements by long-short term memory network," *Biomedical Signal Processing and Control*, vol. 59, p. 101774, 2020.

[8] W. Guo, C. Ma, Z. Wang, H. Zhang, D. Farina, N. Jiang, and C. Lin, "Long exposure convolutional memory network for accurate estimation of finger kinematics from surface electromyographic signals," *Journal of Neural Engineering*, vol. 18, no. 2, p. 026027, 2021.

[9] X. Lv, C. Dai, H. Liu, Y. Tian, L. Chen, Y. Lang, R. Tang, and J. He, "Gesture recognition based on semg using multi-attention mechanism for remote control," *Neural Computing and Applications*, vol. 35, no. 19, pp. 13839–13849, 2023.

[10] C. Chen, W. Guo, C. Ma, Y. Yang, Z. Wang, and C. Lin, "semg-based continuous estimation of finger kinematics via large-scale temporal convolutional network," *Applied Sciences*, vol. 11, no. 10, p. 4678, 2021.

[11] C. Lin, X. Chen, W. Guo, N. Jiang, D. Farina, and J. Su, "A bert based method for continuous estimation of cross-subject hand kinematics from surface electromyographic signals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 87–96, 2023.

[12] Y. Liu, X. Li, L. Yang, G. Bian, and H. Yu, "A cnn-transformer hybrid recognition approach for semg-based dynamic gesture prediction," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–16, 2023.

[13] A. Burrello, F. B. Morghet, M. Scherer, S. Benatti, L. Benini, E. Macii, M. Poncino, and D. J. Pagliari, "Bioformers: Embedding transformers for ultra-low power semg-based gesture recognition," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1443–1448, IEEE, 2022.

[14] M. Montazerin, S. Zabihi, E. Rahimian, A. Mohammadi, and F. Naderkhani, "Vit-hgr: Vision transformer-based hand gesture recognition from high density surface emg signals," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 5115–5119, IEEE, 2022.

[15] M. Montazerin, E. Rahimian, F. Naderkhani, S. F. Atashzar, S. Yanushkevich, and A. Mohammadi, "Transformer-based hand gesture recognition from instantaneous to fused neural decomposition of high-density emg signals," *Scientific Reports*, vol. 13, no. 1, p. 11000, 2023.

[16] I. Tenney, D. Das, and E. Pavlick, "Bert rediscovers the classical nlp pipeline," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, 2019.

[17] Y. Long, Y. Geng, C. Dai, and G. Li, "A transfer learning based cross-subject generic model for continuous estimation of finger joint angles from a new user," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 4, pp. 1914–1925, 2023.

[18] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 894–903, 2017.

[19] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, no. 1, pp. 1–13, 2014.

[20] D. Farina, A. Mohammadi, T. Adali, N. V. Thakor, and K. N. Plataniotis, "Signal processing for neurorehabilitation and assistive technologies," *IEEE Signal Processing Magazine*, vol. 38, no. 4, pp. 5–7, 2021.

[21] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Xceptiontime: independent time-window xceptiontime architecture for hand gesture classification," in *ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1304–1308, IEEE, 2020.

[22] Y. F. Dafalias and M. Taiebat, "Sanisand-z: zero elastic range sand plasticity model," *Géotechnique*, vol. 66, no. 12, pp. 999–1013, 2016.

[23] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[25] Q. Zhang and Y.-B. Yang, "Rest: An efficient transformer for visual recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15475–15485, 2021.

[26] M. Chen, J. Gao, and W. Yu, "Lightweight and optimization acceleration methods for vision transformer: A review," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2154–2160, IEEE, 2022.

[27] H. He, J. Cai, J. Liu, Z. Pan, J. Zhang, D. Tao, and B. Zhuang, "Pruning self-attentions into convolutional layers in single path," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[28] L. Muda, B. KM, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *Journal of Computing*, vol. 2, no. 3, pp. 138–143, 2010.

[29] N. Ketkar, J. Moolayil, N. Ketkar, and J. Moolayil, "Introduction to pytorch," *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch*, pp. 27–91, 2021.

[30] A. Matran-Fernandez, I. J. Rodríguez Martínez, R. Poli, C. Cipriani, and L. Citi, "Seeds, simultaneous recordings of high-density emg and finger joint angles during multiple hand movements," *Scientific data*, vol. 6, no. 1, p. 186, 2019.

[31] C. Lin, X. Niu, J. Zhang, and X. Fu, "Improving motion intention recognition for trans-radial amputees based on semg and transfer learning," *Applied Sciences*, vol. 13, no. 19, 2023.

[32] D. Zheng, Y. Liu, L. Li, *et al.*, "Leveraging inter-layer dependency for post-training quantization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6666–6679, 2022.