TraHGR: Transformer for Hand Gesture Recognition via Electromyography

Soheil Zabihi, Elahe Rahimian, Amir Asif, Senior Member, IEEE, and Arash Mohammadi, Senior Member, IEEE

Abstract—Deep learning-based Hand Gesture Recognition (HGR) via surface Electromyogram (sEMG) signals have recently shown considerable potential for development of advanced myoelectric-controlled prosthesis. Although deep learning techniques can improve HGR accuracy compared to their classical counterparts, classifying hand movements based on sparse multichannel sEMG signals is still a challenging task. Furthermore, existing deep learning approaches typically include only one model as such can hardly extract representative features. In this paper, we aim to address this challenge by capitalizing on the recent advances in hybrid models and transformers. In other words, we propose a hybrid framework based on the transformer architecture, which is a relatively new and revolutionizing deep learning model. The proposed hybrid architecture, referred to as the Transformer for Hand Gesture Recognition (TraHGR), consists of two parallel paths followed by a linear layer that acts as a fusion center to integrate the advantage of each module. We evaluated the proposed architecture TraHGR based on the commonly used second Ninapro dataset, referred to as the DB2. The sEMG signals in the DB2 dataset are measured in real-life conditions from 40 healthy users, each performing 49 gestures. We have conducted an extensive set of experiments to test and validate the proposed TraHGR architecture, and compare its achievable accuracy with several recently proposed HGR classification algorithms over the same dataset. We have also compared the results of the proposed TraHGR architecture with each individual path and demonstrated the distinguishing power of the proposed hybrid architecture. The recognition accuracies of the proposed TraHGR architecture for the window of size 200ms and step size of 100ms are 86.00%, 88.72%, 81.27%, and 93.74%, which are 2.30%, 4.93%, 8.65%, and 4.20% higher than the state-of-the-art performance for DB2 (49 gestures), DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures), respectively.

I. INTRODUCTION

To improve the quality of life of people with upper limb amputation, recently, there has been a growing interest in development of learning-based myoelectric prostheses using multi-channel surface Electromyogram (sEMG) signals [1], [2], [3]. The information obtained from the sEMG signals, which are related to the neural activities of the underlying muscles, is used to decode the movements of the targeted limb. Generally speaking, sEMG signals can be collected via two different recording techniques, i.e., sparse multi-channel sEMG, and High-Density sEMG (HD-sEMG). The former, typically, consists of a limited number of electrodes with sparse placement, which is the commonly used sEMG recording modality in wearable systems [4], [5]. On the other hand, an HD-sEMG device consists of a grid of electrodes that collect information about the temporal and spatial electrical activities of the underlying muscles enabling them to capture large amounts of data [5], [6], [7]. This, however, results in increased complexity of the underlying system, which in turn challenges their ease of applicability in wearable systems and adds latency to the processing pipeline. Recent studies [8], therefore, focus on the use of sparse multi-channel sEMG devices given their ease of use and reduced complexity [9]. The use of sparse multi-channel sEMG datasets can, however, challenge the gesture recognition performance due to its sensitivity to the electrode location. In particular, while Deep Neural Networks (DNNs) achieve high performance in gesture recognition using HD-EMG devices, their efficacy is limited for sparse multi-channel sEMG devices in which a shallow dataset is collected with a lower sampling rate and limited number of electrodes. Inspired by the above discussion, as well as the significant gap between the performance of existing methods for HD-EMG and the sparse multi-channel sEMG approaches, the paper aims to present a new hybrid learning framework for achieving superior performance using sparse multi-channel signals. Specifically, the primary goal of this study is to investigate the potential of transformer-based hybrid models to achieve high accuracy for hand gesture recognition based on sEMG signals.

Manuscript received 28 March 2022; revised 30 December 2022 and 10 July 2023; accepted 6 October 2023. Date of publication 13 October 2023; date of current version 27 October 2023. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada through the NSERC Discovery under Grant RGPIN-2023-05654. (Corresponding author: Arash Mohammadi.)

Soheil Zabihi is with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: s_zab@encs.concordia.ca).

Elahe Rahimian and Arash Mohammadi are with the Concordia Institute for Information System Engineering (CIISE), Concordia University, Montreal, QC H3G 2W1, Canada (e-mail: e_ahimia@encs.concordia.ca; arash.mohammadi@concordia.ca).

Amir Asif is with the Department of Electrical Engineering and Computer Science, York University, Toronto, ON M3J 1P3, Canada (e-mail: asif@eecs.yorku.ca).

Digital Object Identifier 10.1109/TNSRE.2023.3324252

© 2023 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Contributions: In most of the recent studies for sEMG-based hand recognition, the existing deep learning approaches involve only a single model, which can barely capable to extract more generic representations in the single network structures. The paper addresses this gap by designing a Transformer-based hybrid solution that has great potential for extracting temporal and spatio-temporal representation to improve HGR accuracy. In particular, capitalizing on the recent success of Transformers in various fields of Machine Learning [10], [11], [12], [13], we aim to examine its applicability and potential for sEMG-based hand gesture recognition. The proposed method, referred to as the Transformer for Hand Gesture Recognition (TraHGR), increases the accuracy of sEMG decoding for the classification of hand movements.

The proposed framework consists of two parallel paths (one Temporal transformer Network (TNet) and one Feature transformer Network (FNet)) followed by linear layers, which integrate the output of paths to provide better representation for different numbers of hand movement recognition. The TNet is used to extract temporal features while simultaneously the FNet is utilized to extract spatio-temporal features, which are then fused to augment the discriminating power of the model leading to improvement of the overall performance of the HGR classification task. Performance of the proposed TraHGR framework is evaluated using the second and fifth Ninapro dataset [14], [15], [16], [17], which are widely used datasets providing sparse multi-channel sEMG signals from diverse hand movements similar to those obtained in real-life conditions. As a result, Ninapro dataset enables development of innovative DNN-based recognition solutions for the HGR tasks. We have conducted an extensive set of experiments to test and validate the proposed TraHGR architecture, and compare its achievable accuracy with several recently proposed HGR classification algorithms based on the same datasets. Results show that the proposed TraHGR outperforms all existing solutions over the DB2 dataset and its sub-exercises. In summary, the contributions of the paper are as follows:

- The paper for the first time, to the best of our knowledge, develops a hybrid Transformer based architecture for the task of HGR via sparse multi-channel sEMG.
- The paper demonstrates the superior performance of the proposed hybrid architecture TraHGR and its ability to extract more distinctive information for gesture recognition over the single models; i.e., TNet and FNet.
- The paper investigates the impact of various window sizes, i.e., 200ms, 150ms, and 100ms, on both the overall performance and model complexity.
- The paper classifies a high number (49) of gestures with a high accuracy. More specifically, compared with the proposed architectures in the recent state-of-the-art studies, TraHGR improves the recognition accuracies to 86.18% on DB2 (49 gestures), to 88.91% on the DB2-B (17 gestures), to 81.44% on DB2-C (23 gestures), and to 93.84% on the DB2-D (9 gestures).

The rest of the paper is organized as follows: In Section II, an overview of related works is provided. Section III describes the database and pre-processing step, we also present details of the proposed TraHGR architecture in this section. The experiments and results are presented in Section IV. Finally, the conclusion is presented in Section V.

II. RELATED WORKS

Deep MyoLearning can be categorized into regression-based techniques and classification-based techniques. Regression-based techniques (typically, via a combination of several regressors) have been used to identify and estimate different movements in a continuous space across several degree-of-freedoms (DoFs). Unfortunately, the prediction accuracy of such techniques is not yet at a level to be used in practical settings. As an alternative, some researchers such as [18] adopt a regression Convolutional Neural Network (CNNs) to estimate wrist motions based on sEMG signals. Regression CNNs allow for independent and simultaneous control so that several DoFs can be manipulated concurrently with different magnitudes. This allows for more dextrous and realistic prosthetic motion than low-DoF proportional control or discrete classification control. This CNN-based regression technique was validated by real-time control tests that demonstrated superior performance compared to SVM-based techniques [18]. In [19], the grasping force levels based on a combination of Principal Component Analysis (PCA) and DNN are predicted for the control of a prosthetic hand. More specifically, dimension reduction of time domain features are achieved by PCA, and then an sEMG-force regression model is generated by using DNN. This is an important step toward improving the grip of prosthetic hand based on force prediction.

Generally speaking, the developed methods for classifying hand movements can be categorized into the following three main categories: (i) Traditional approaches based on classic Machine Learning (ML) methods [17], [20], [21], [22], [23], (ii) DNN-based techniques [5], [23], [24], [25], [26], [27], [28], [29], and (iii) Hybrid methodologies [6] that combine multiple models. The existing researches on prosthetic myoelectric control focus primarily on traditional ML approaches as a common strategy for HGR task [30]. In such methods, handcrafted features, in time domain, frequency domain, or time-frequency domain [6], are first extracted by human experts, which are then fed to a ML classifier. Extraction and feature selection, however, can affect the overall performance [31], as such some researchers [5] have explored and integrated several classical feature sets that provide multi-view of sEMG signals to achieve higher gesture recognition accuracy. On the other hand, different classifiers such as Discriminant Analysis (LDA), Support Vector Machine (SVM), and Random Forests (RF), and Principal Components Analysis (PCA) are utilized in the literature [14], [17], [23], [31], [32] to increase the discriminating power of the model and improve gesture recognition performance.

Although the traditional ML-based approaches have shown strong potential for HGR task, more recently, there has been a great deal of interest in using deep-learning architectures to process multi-channel sEMG signals and increase the discrimination power of the model. In particular, it has been shown [23] that the automatic feature extraction used in deep learning architecture can lead to higher classification.
accuracy. More specifically, the authors in [23], for the first time, used the Convolutional Neural Network (CNN) architecture to classify hand gestures, which showed its potential to improve the overall performance compared to existing traditional approaches. This achievement was the starting point for considering CNN as a promising approach in the context of sEMG data classification [26], [29], [33]. For example, in Reference [7], the authors proposed and used the CNN architecture to extract spatial information from sEMG signals and perform HGR classification. In addition to CNN-based architectures, some researches [34], [35] used Recurrent Neural Networks (RNNs) to extract the temporal features from the sEMG signals. RNNs are used because sEMG signals are sequential in the nature, and recurrent-based networks such as Long Short Term Memory (LSTM) can extract the patterns in a sequence of sEMG data treating HGR as a sequence modeling task. In addition, it has been shown [36] that the proper design of CNN architectures can outperform RNNs in sequence modeling. In this respect, some researchers [8], [27], [28] used temporal convolutions for HGR and showed its potentials to extract temporal features.

Alternatively, hybrid architectures such as CNN-RNN have shown promising results in classifying hand movements [6], [25], as they benefit from advantages of different modules in extracting temporal and spatial features. Meanwhile, with the advent of the attention mechanism [10], transformers are being considered as a new ML technique for sequential data modeling [37], [38]. Capitalizing on the recent success of transformers in various fields such as machine translation [11], [39], speech recognition [40] and computer vision [12], we aim to examine its applicability and potentials for sEMG-based gesture recognition. In other words, we recognized an urgent need to develop a transformer-based hybrid architecture to augment the recognition accuracy of HGR. In this paper, we introduce the TraHGR architecture, which increases the accuracy of sEMG decoding in the classification of human hand movements. In addition, we examine the complexity and performance of different types of TraHGR architectures.

We would like to point out that one inherent problem in the sEMG-based hand gesture recognition task is the time- and user-dependent nature of the sEMG signal [41]. In other words, due to physiological differences in muscle activities from one user to another, a pre-trained model on existing users (source) cannot be expected to perform well on a new user (target) [42], [43]. In addition, various electrophysiological and user-related variables can affect sEMG signals. These include muscle fatigue [44], changes in electrode-skin impedance due to perspiration/humidity [45], electrode displacement [46], and user-related issues such as variations in contraction intensity, hand orientations, and arm postures [47]. As a result of these changes, the accuracy of a pre-trained model on source data may degrade when testing on the target user due to the domain shift. To this end, domain adaptation methods are highly recommended in this field of study, where learning algorithms involve some techniques to transfer information from the source to the target domain despite the existence of a distribution mismatch among them. As a result, Transfer Learning (TL) techniques in upper-limb hand motion estimates have received a lot of attention in recent years [48], [49], [50], [51], [52], [53]. Furthermore, in [8], as a domain adaptation methodology, a novel Few-Shot learning method is introduced with the objective of inferring the desired output based on just one or a few observations from the target domain, resulting in a promising performance for unseen user hand gesture recognition. However, in this paper, the primary goal is to investigate the potential of a transformer-based hybrid solution for offline hand gesture recognition based on sEMG signals. Therefore, following the previous studies [5], [6], [14], [23], [32], [33], training of the TraHGR network is conducted in a supervised fashion for each user independently. Although we conducted an experiment to investigate the impact of a simple TL technique on the performance of the proposed model, investigating domain adaptation techniques is not the primary goal of the study. Moreover, as mentioned earlier, we have an offline setting in this study. From offline gesture recognition in experiment settings to the real-time clinical environment, the reported accuracy might drop [54]. These may be seen as both a constraint for the current study and a promising area for future work.

III. MATERIAL AND METHODS

A. Database

The proposed TraHGR architecture is evaluated on the second Ninapro database [14], [15], [16], which is a publicly available dataset for sEMG-based human-machine interfacing. The second database Ninapro, which is referred to as the DB2, was collected from 40 users. Each user performs 49 movements in which each movement is repeated 6 times, each time lasting for 5 seconds, followed by 3 seconds of rest. The sEMG signals were gathered using Delsys Trigno Wireless EMG system with 12 wireless electrodes, sampled at 2 kHz. The DB2 dataset was presented in three exercises B, C, and D, which consist of different types of movements. In particular, Exercise B, C, and D consists of 17, 23, and 9 movements, respectively. For the rest of this paper, Exercise B, C, and D are referenced to DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures), respectively. Following the recommendations provided by Ninapro benchmark datasets [14], [15], [16], we consider the repetitions 2 and 5 of each movement as the holdout test set, and the remaining four repetitions, i.e., 1, 3, 4, and 6, as the training set. However, since the benchmark does not clearly specify the validation set, the first, second, third, and fourth quarters of repetitions 1, 3, 4, and 6, respectively, are used in this study as the validation set.

B. Pre-Processing Step

The EMG signals are pre-processed for classification purposes before being fed into the proposed architecture. The pre-processing phase consists of three steps, i.e., low-pass filtering, normalization, and segmentation. More specifically, we followed the procedure outlined in previous studies [7], [23], [29] and used the low-pass Butterworth filter. To enhance the performance of the proposed architecture, we applied the low-pass filter three times with different order of filters.
namely 1, 3, and 5, and then concatenated all filtered signals together to form three-channel sEMG signals. Then, for the normalization step, we are inspired by the $\mu$-law normalization technique introduced in [26] and [8], which is defined as follows

$$F(x_t) = \text{sign}(x_t) \frac{\ln (1 + \mu |x_t|)}{\ln (1 + \mu)},$$

where the input scaler is represented by $x_t$, and the new range of the signal is indicated by parameter $\mu$. After normalization, the sEMG signals are segmented with a sliding window. Each input from the sEMG signal segmentation phase is denoted by $X \in \mathbb{R}^{S \times W \times C}$, where $S$ shows the number of sensors in the DB2 dataset, $W$ shows the number of samples of electrical activities of muscles obtained at the rate of 2 kHz for a window of 200ms, 150ms, or 100ms, and $C$ denotes the number of channels of the sEMG signals.

The acceptable amount of delay in sEMG-based models can vary depending on the specific application and user requirements. While earlier literature has traditionally considered a 300ms threshold as an acceptable delay [55], recent research has suggested that the optimal controller delay should ideally be below 175ms, with preferred values even below 125ms [56]. These values are based on systematic investigations into the optimal delay for improved controller performance. However, achieving low delays in sEMG-based models can be challenging due to the processing and computational requirements involved. As a result, researchers have explored techniques such as overlapping windows to mitigate the delay. In this study, we considered window sizes of 150ms and 100ms in addition to the commonly used 200ms window size to ensure fair comparisons. The step size of the sliding window is set to 10ms for all experiments, results for a step size of 100ms are included to allow for a fair comparison with other studies.

C. Overview of the TraHGR Architecture

The TraHGR architectural design is based on transformers in which the attention mechanism is the main building block. The attention mechanism has been used in previous studies [6], [8], [27] along with CNNs and/or recurrent-based architectures for HGR task. However, in this paper, we show that transformer-based architectures that rely solely on attention mechanisms can perform better than previous studies in which hybrid architectures (e.g., attention integrated with CNN or RNN) have been adopted. The overall proposed architecture is illustrated in Fig. 1, which is inspired by the Vision Transformer (ViT) [12], in which each input is divided into patches, and the network is supposed to perform label prediction based on the sequence of these patches. As shown in Fig. 1, the proposed TraHGR consists of a TNet path implemented in parallel with a FNet path followed by a linear layer, which acts as the fusion centre combining the extracted features from each of the two parallel paths in order to classify the hand gestures. In the following subsections, we will further elaborate on the details of the proposed architecture.

D. Embedded Patches

In this sub-section, we focus on the input of the transformer encoder, which is a sequence of embedded patches. As illustrated in Fig. 1, the embedded patches are constructed from patch embeddings and position embeddings, which are described below.

1) Patch Embeddings: As mentioned earlier, we split each segment of sEMG signals $X$ into non-overlapping patches $X_p = \{x_p\}_{i=1}^{N}$. More specifically, each segment $X \in \mathbb{R}^{S \times W \times C}$ is divided into $N$ non-overlapping patches in which each patch is flattened. We represented the sequence of these flatten patches with $X_p \in \mathbb{R}^{N \times (P_1 \times P_2 \times C)}$, where $C$ denotes the number of channels, $(P_1, P_2)$ shows the size of each patch, and $N = S \cdot W / P_1 \cdot P_2$ represents the length of this sequence, i.e., the number of patches. As shown in Fig. 1, we applied two types of patching:

- **Temporal Patching**: Here, the size of each patch is $(1, W)$; therefore, the number of patches is $N = S$. This type of patching is called Temporal Patching because each patch contains the information from only one of the sensors in the dataset for a sequence with length $W$. The TNet path is designed in such a way that they can take into account the temporal patches as the input.

- **Featural Patching**: We set the size of each patch to $(S, S)$, i.e., $P_1 = P_2 = S$, therefore, the number of patches is $N = W / S$. We refer to this type of patching as Featural because each patch contains the information of all $S$ sensors for a sequence with length $S$. Therefore, both spatial and temporal information are included in a Featural patch. The Featural patches are provided as the input only to the FNet layer as shown in Fig. 1.

Finally, a linear mapping is introduced to create the embedding for each of these patches (Fig. 1). More specifically, a matrix $E \in \mathbb{R}^{(P_1 \cdot P_2 \cdot C) \times D}$ is shared among different patches to linearly project each patch into the model dimension $D$ (Eq. (2)). The output of this projection is known as the Patch Embeddings.

2) Class Token: Similar to the Bert framework [11], a trainable embedding is prepended to the sequence of patch embeddings $(Z_0^0 = x_{\text{class}})$ with the goal of capturing the meaning of the entire segmented input as a whole. More specifically, the class token’s embedding after the last transformer encoder layer $(Z_1^L)$ is used for classification purposes (Eq. (10)).

3) Position Embeddings: As HGR based on sEMG signals is a time-series processing task, the order of data is an essential part for sequence modeling. Recurrent-based architectures such as LSTM inherently consider signal order, however, transformers do not process the input sequentially and combine the information of all the elements through attention mechanism. Therefore, there is a need to encode the order of each element in the sequence. This is where positional embedding comes in. In fact, position embedding allows the network to determine where a particular patch came from. There are several ways to retain position information at the transformer input, e.g., Sinusoidal positional embedding, 1-dimensional positional embedding, 2-dimensional positional, and Relative positional embeddings embedding [12], [40]. Following [12], we used the standard trainable 1-dimensional positional embeddings. As shown in Fig. 1, position embeddings indicated by
The proposed TraHGR architecture consists of two parallel paths (one TNet and one FNet). Each segment of sEMG signals \( X \) is divided into \( N \) non-overlapping patches. The TraHGR uses the TNet path to get the temporal patches while simultaneously the FNet is utilized to consider the featural patches. In both TNet and FNet, the patches are mapped linearly into the model dimension \( D \). We refer to the output of this step as “Patch Embedding”. Then, a “class token” is prepended to the sequence of patch embeddings which is finally used for the classification purpose.

The “Positional Embedding” is added to the “Patch Embedding” to retain the positional information. The output of this step is called “Embedded Patches” and is fed to the Transformer encoder consisting of \( L \) layers, each layer consisting of MSA and MLP modules. Finally, we add the output of the TNet and FNet class tokens to get the final representation, which then acts as the input to the linear layer.

\( \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \) is added to the patch embeddings. The formulation which governs patch and position embeddings is as follows

\[
\mathbf{Z}_0 = [x_{\text{class}}; x_1^p \mathbf{E}; x_2^p \mathbf{E}; \ldots; x_N^p \mathbf{E}] + \mathbf{E}_{pos}.
\]

The output of Eq. (2) is called Embedded Patches, which are fed as an input to the transformer encoder.

### E. Transformer Encoder

The transformer encoder takes the \( \mathbf{Z}_0 \) as an input. This block is inspired from the main transformer encoder introduced in [10], which treats all embedded patches as tokens. As illustrated in Fig. 1, the transformer encoder consists of \( L \) layers. Each layer contains two modules, namely the Multihead Self-Attention (MSA) and a Multi-Layer Perceptron (MLP) module, i.e.,

\[
\mathbf{Z}_l' = \text{MSA}(\text{LayerNorm}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1}, \quad l = 1 \ldots L
\]

\[
\mathbf{Z}_l = \text{MLP}(\text{LayerNorm}(\mathbf{Z}_l')) + \mathbf{Z}_l', \quad l = 1 \ldots L
\]

It is worth noting that a layer-normalization [57] is used before MSA and MLP modules, and the residual connections are applied to address degradation problem. The MLP module consists of two linear layers in which the first layer is followed by Gaussian Error Linear Unit (GELU) activation function. Moreover, the MSA module is defined based on the Self-Attention (SA) mechanism, which is discussed next.

1) Self-Attention (SA): The SA mechanism [10] measures the pairwise similarity of each query and all the keys and obtains a weight for each value. Finally, the output is computed based on the weighted sum over of all values. In particular, if we define an input \( \mathbf{Z} \in \mathbb{R}^{N \times D} \) consisting of \( N \) vectors, each of length \( D \), the three matrices, i.e., Queries \( \mathbf{Q} \), Keys \( \mathbf{K} \), and Values \( \mathbf{V} \), are calculated as follows

\[
[\mathbf{Q}, \mathbf{K}, \mathbf{V}] = \mathbf{W}_{QKV},
\]

where \( \mathbf{W}_{QKV} \in \mathbb{R}^{D \times 3D_h} \) denotes the trainable weight matrix and \( D_h \) shows the length of each vector in \( \mathbf{Q} \), \( \mathbf{K} \), and \( \mathbf{V} \). To measure the weights for \( \mathbf{V} \), the dot-product of \( \mathbf{Q} \) and \( \mathbf{K} \) is
calculated, then scaled with $\sqrt{D_h}$. These weights are converted to the probabilities $P \in \mathbb{R}^{N \times N}$ using the softmax function as follows

$$P = \text{softmax}(\frac{QK^T}{\sqrt{D_h}}).$$

(6)

Finally, the output of SA mechanism is computed as follows

$$SA(Z) = PV.$$  

(7)

By using the attention mechanism, the model pinpoints a specific information in the input sequence.

2) Multihead Self-Attention (MSA): Here, the SA mechanism is used for $h$ times in parallel, allowing the architecture to pinpoint specific pieces of information in the input sequence for each head differently. In particular, each head has its own trainable weight matrix. The final matrix in the MSA mechanism is a projection of the concatenated outputs of the $h$ heads, which is formulated as follows

$$MSA(Z) = [SA_1(Z); SA_2(Z); \ldots; SA_h(Z)]W_{MSA}.$$  

(8)

where $W_{MSA} \in \mathbb{R}^{h \times D_h \times D}$. Here, $D_h$ is set to $D/h$ to keep the number of parameters constant when $h$ changes.

F. TraHGR’s Output

As shown in Fig. 1, the TraHGR consists of two paths, i.e., TNet and FNet. For each path, the aforementioned calculations (Eqs. (2)-(8)) are performed in parallel. Then, the predicted class labels of each path is calculated based on its corresponding $Z_l^0$ as follows

$$y_{path} = \text{Linear}(\text{LayerNorm}(Z_l^0)_{path}),$$

(9)

where $path \in \{\text{TNet}, \text{FNet}\}$. Finally, the output of the TraHGR is calculated based on the sum of $Z_l^0$ in the TNet and FNet as follows

$$y = \text{Linear}(\text{LayerNorm}([Z_l^0]_{\text{TNet}} + [Z_l^0]_{\text{FNet}})).$$

(10)

It is worth mentioning that $y_{\text{TNet}}, y_{\text{FNet}}$, and $y$ are used for TraHGR training. More details are provided in the subsection III-G. This completes description of the proposed TraHGR architecture, next, its performance is evaluated through several experiments.

G. Loss Function

The loss function $L$ of the TraHGR consists of the following three components

$$L = L_{\text{TNet}} + L_{\text{FNet}} + L_{\text{TraHGR}},$$

(11)

where the first term $L_{\text{TNet}}$ is loss of the TNet path in the proposed TraHGR architecture. More specifically, cross-entropy loss is considered for measuring classification performance using the TNet’s output $y_{\text{TNet}}$ (Eq. (9)) and the target values. Similarly, the second term $L_{\text{FNet}}$ is the cross-entropy loss computed using the second path (FNet) of the TraHGR architecture where FNet’s outputs $y_{\text{FNet}}$ (Eq. (9)) are considered. Finally, the last term $L_{\text{TraHGR}}$ is calculated using the TraHGR’s output $y$ (Eq. (10)).

H. Different Variants of TraHGR

TABLE I shows different configurations of the hyperparameters in the TraHGR architecture resulting in different variants of the model denoted by TraHGR-Base, TraHGR-Large, and TraHGR-Huge. The introduction of different variants of the model, namely TraHGR-Base, TraHGR-Large, and TraHGR-Huge, in our study, serves the purpose of exploring the impact of various hyperparameter configurations on the performance of the model. By investigating these variants, we can gain insights into how different choices of hyperparameters, such as the number of layers ($l$), model dimension ($D$), MLP size, and the number of heads ($h$), influence the accuracy of the model. This analysis allows us to understand the trade-off between model complexity and performance. Moreover, the single deep networks TNet and FNet and their corresponding huge version were designed and trained as individual models. The goal of designing and training these single networks is to evaluate their performance and compare it against the performance of the hybrid models in order to assess the effectiveness of the TraHGR architecture in achieving better results. More precisely, the comparison provides insights into the effectiveness and advantages of the proposed hybrid approach for representative feature extraction and classification tasks. In TABLE I, the number of parameters (Params) is calculated for DB2 (49 gestures) while this number will be less for DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures).

IV. EXPERIMENTS AND RESULTS

In this section, we evaluate performance of the proposed TraHGR architecture through a series of experiments. In all experiments, the Adam optimizer [58] was used with learning rate of 0.0001 and the weight decay of 0.001. Moreover, the batch size is set to 512. Different variants of TraHGR, TNet, and FNet, provided in TABLE I, are used for training and evaluation purposes with window size $\in \{200ms, 150ms, 100ms\}$.

A. Evaluation of the Proposed TraHGR Architecture

This subsection provides evaluations on the prediction performance of the proposed hybrid transformer-based architecture. In this regard, first, we compare different variants of the TraHGR architecture and show the effect of different hyperparameters (e.g., number of layers, model dimension, MLP size, and number of heads) on the overall accuracy. Then, to demonstrate the performance of the hybrid transformer, we also compare the TraHGR architecture with single deep models, i.e., TNet and FNet and their huge version.

TABLE II, III, and IV show HGR recognition accuracy, which is averaged over all subjects for the test set. From TABLE II, it can be observed that the proposed TraHGR-Huge architecture outperformed other TraHGR architecture variants (TraHGR-Base and TraHGR-Large) when evaluated based on the DB2 (49 gestures) for the same window size. However, as shown in TABLE I, the number of parameters of the TraHGR-Huge is much higher than that of the TraHGR-Base and TraHGR-Large models. This fact indicates that increasing the number of layers ($l$), model dimension ($D$), MLP size,
and number of heads \((h)\) have a positive effect on the model's accuracy, however, this comes with the cost of increasing the complexity. In addition, as shown in TABLE I, each model has a larger number of trainable parameters for window size 200ms than its counterpart in the window size of 150ms or 100ms, resulting in higher complexity. However, as shown in TABLE II, a larger window size can further improve the results because the model has access to a longer sequence length.

### B. TraHGR Hybrid Architecture Versus TNet and FNet

We independently trained and evaluated the proposed model on DB2 subsets, i.e., DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures). In Fig. 2, the performance of the proposed architectures for DB2 (49 gestures) and its three sub exercises, i.e., B, C, and D are shown. It can be observed that for both window sizes of 200ms and 150ms achieving a high accuracy for DB2-C is more challenging than DB2-B and DB2-D subsets. More specifically, DB2-C consists of 23 grasping and functional movements for which everyday objects like bottle and knife are presented to the user for grasping, in order to mimic daily-life actions such as opening a bottle or cutting something [14]. Therefore, the performance reduction in the DB2-C subset is not far from expectation as the muscle groups which are predominantly used during movements of DB2-C are more complicated than basic hand posture and wrist movements for which everyday objects like bottle and knife are presented to the user for grasping.

### TABLE I

| Model          | #Layers \((L)\) | Model dimension \((D)\) | MLP size \(200\text{ms}\) | #Heads \((h)\) | #Params \(100\text{ms}\) |
|----------------|----------------|-------------------------|---------------------------|-------------|-------------------|
| TraHGR-Base    | 1              | 32                      | 128                       | 4           | 83.731            |
| TraHGR-Large   | 2              | 64                      | 256                       | 4           | 316.051           |
| TraHGR-Huge    | 1              | 144                     | 720                       | 8           | 846.579           |
| TNet           | 1              | 144                     | 720                       | 8           | 472.513           |
| FNet           | 1              | 144                     | 720                       | 8           | 366.673           |
| TNet-Huge      | 1              | 200                     | 1084                      | 8           | 846.733           |
| FNet-Huge      | 1              | 224                     | 1176                      | 8           | 846.377           |

### TABLE II

| Model          | Accuracy \(\pm\) STD \(200\text{ms}\) | Accuracy \(\pm\) STD \(150\text{ms}\) | Accuracy \(\pm\) STD \(100\text{ms}\) |
|----------------|----------------------------------------|----------------------------------------|----------------------------------------|
| TraHGR-Base    | 78.60 \(\pm\) 6.03                    | 77.54 \(\pm\) 5.99                    | 76.17 \(\pm\) 6.09                    |
| TraHGR-Large   | 83.58 \(\pm\) 5.48                    | 82.58 \(\pm\) 5.60                    | 81.30 \(\pm\) 5.87                    |
| TraHGR-Huge    | 86.18 \(\pm\) 4.99                    | 85.43 \(\pm\) 5.24                    | 84.13 \(\pm\) 5.21                    |

### TABLE III

| Model          | Accuracy \(\pm\) STD \(200\text{ms}\) | Accuracy \(\pm\) STD \(150\text{ms}\) | Accuracy \(\pm\) STD \(100\text{ms}\) |
|----------------|----------------------------------------|----------------------------------------|----------------------------------------|
| TraHGR-Huge    | 86.18 \(\pm\) 4.99                    | 85.43 \(\pm\) 5.24                    | 84.13 \(\pm\) 5.21                    |
| TNet           | 83.39 \(\pm\) 5.44                    | 82.81 \(\pm\) 5.60                    | 81.43 \(\pm\) 5.88                    |
| FNet           | 80.72 \(\pm\) 5.82                    | 80.05 \(\pm\) 6.03                    | 79.38 \(\pm\) 6.15                    |
sizes 200ms and 150ms, TNet has approximately 1.5× more parameters than TraHGR-Large, and FNet is almost 1.2× larger. For the window size of 100ms, both single networks have almost 1.3× more parameters than TraHGR-Large. However, as shown in TABLE II and III, TraHGR-Large has comparable performance to TNet, and it outperforms FNet, showing that a hybrid model with fewer number of parameters is capable to extract more generic representations resulting in comparable or even better performance compared to larger single networks, TNet and FNet. According to TABLE I, the network structure in TNet and FNet is completely different than TraHGR-Large. Therefore, we conducted a new experiment in which the structure of the single and hybrid networks remained unchanged. To do so, we can compare the performance of TraHGR-Huge against the TNet and FNet (see TABLE I). As shown in TABLE III, the TraHGR-Huge outperforms the single deep models (TNet and FNet) when the structure of the networks is preserved. However, since the number of trainable parameters is TraHGR-Huge is considerably larger than TNet and FNet, the performance improvement could be conducted due to the TraHGR-Huge capacity to represent more complex hypothesis space. As a result, we conducted new experiments in which the number of parameters for new variants of TNet and FNet architectures is expanded to be on the same scale as TraHGR-Huge. Specifically, to increase the number of parameters in new variants of TNet and FNet, we began by increasing model dimension $D$ and stopped just before exceeding the number of parameters in TraHGR-Huge. Then, the size of the MLP layer in the transformer encoder is enlarged to fill the remaining gap in terms of the number of parameters as much as possible, resulting in TNet-Huge and FNet-Huge architectures. Detailed information about the structure of different variants of these single networks and their number of parameters are provided in TABLE I. As shown in TABLE IV, TraHGR-Huge significantly outperforms TNet-Huge and FNet-Huge architectures while they are all in the same scale.

As shown in TABLE III and IV, although the number of trainable parameters in TNet-Huge and FNet-Huge are significantly increased compared to TNet and FNet, their average recognition accuracy improvement for different window sizes is not significant. As a result, since the single networks were not capable to achieve high performance even with massive parameters expansion and given the outstanding performance of TraHGR-Huge architecture, it can be concluded that the hybrid approach integrates the advantages of two parallel paths to model better and more generic representation resulting in performance improvement. It is worth mentioning that for hybrid models such as TraHGR-Base, TraHGR-Large, and TraHGR-Huge, the classification accuracy is calculated using the output of Eq. (10), while for single deep models such as TNet and FNet this number is computed using the output of Eq. (9).

C. Computation Time

The window size of 200ms is a well-established value in previous works [5], [6], [14], [23], [32], [33], [59], [60] in sEMG-based hand gesture recognition. Following these
TABLE V
COMPARING AVERAGE COMPUTATION TIME OF DIFFERENT VARIANTS
OF TRAHRG FOR HAND GESTURE RECOGNITION ACROSS TEST
SETS OF ALL SUBJECTS IN THE DB2 (49 GESTURES) DATASET
FOR DIFFERENT VARIANTS OF TRAHRG ARCHITECTURE ON
SEVERAL WINDOW SIZES (200ms, 150ms, AND 100ms)

| Model       | Average Time (ms) Per-Sample |
|-------------|-----------------------------|
|             | 200ms | 150ms | 100ms |
| TraHGR-Base | 0.113 | 0.083 | 0.042 |
| TraHGR-Large| 0.207 | 0.123 | 0.072 |
| TraHGR-Huge | 0.241 | 0.150 | 0.095 |

D. Statistical Analysis

Following [8], [53], we considered each user as a separate dataset and conduct Wilcoxon signed-rank test [61] and Bonferroni correction to ensure the reliability and validity of our results. To do so, given that we have 40 users, for each model we will have 40 accuracies resulting from each user’s test set. Having accuracies for each model, we performed statistical analysis on the effectiveness of the observations for DB2 (49 gestures) for the window size of 200ms. According to the results shown in Fig. 3, the difference in accuracy between TraHGR-Huge and other proposed architectures such as TraHGR-Base, TraHGR-Large, TNet, and FNet, for window sizes 200ms were considered statistically significant by the Wilcoxon signed-rank. In Fig. 3, the p-value of significance is considered 0.05 and the annotated * mark represents \( p \leq 0.05 \).

Worth to mention that, in Fig. 3, the adjusted p-values resulting from the Bonferroni correction are below the threshold of \( 10^{-4} \) (i.e., \( p \leq 10^{-4} \)) for statistically significant comparisons. Furthermore, Figure 3 provides a visual representation of the performance distribution across 40 users for each of the proposed models. Each boxplot shows the Interquartile Range (IQR), which presents the performance of each model for all users into quartiles. More specifically, the upper and lower whiskers show the 75th and 25th percentiles. In a sense that, in each boxplot, the achieved accuracy for 25% of the users, i.e., 10 users, are in the range defined by the lower whisker and the other 25% of the users have accuracy in the range defined by the upper whisker. The horizontal lines at the beginning of the lower whisker and the end of the upper whisker indicate the models’ minimum and maximum accuracies, respectively. Finally, the boxplot covers the range of accuracy for 50% of the users. The horizontal line in each boxplot illustrates the median performance. In a sense that the accuracy of 25% of users falls into the bottom portion of the box, while the other 25% of users fall into the higher part of the box. As shown in Fig. 3, The boxplot corresponding to TraHGR-Huge compared to other counterparts is shifted up. In other words, TraHGR-Huge has improved the performance of all users. Furthermore, when comparing different TraHGR variations, it is clear that increasing the number of parameters led to an increase in accuracy due to the models’ enhanced capacity to extract more generic representations. However, increasing the number of parameters does not have a significant improvement on the TNet and FNet as shown in the Fig. 3, “ns” stands for not significant, i.e., \( 0.05 < p \text{-value} \leq 1 \).

To evaluate the robustness of the proposed approach, we conducted 100 Monte Carlo (MC) runs, where sensor measurements were intentionally contaminated with additive Gaussian noise at predefined signal-to-noise ratio (SNR) levels. Specifically, in each MC run, we introduced noise based on an SNR of 25dB. The MC simulation results, obtained
by averaging the accuracy over the 100 runs, showed that the proposed TraHGR-Huge achieved an accuracy of 85.68% with a standard deviation of 5.32%. Furthermore, we compared these results to the accuracy obtained without MC simulation, which yielded a slightly higher accuracy of 86.18% with a standard deviation of 4.99% (as shown in TABLE VI). These findings highlight the consistent and stable performance of the proposed model, reinforcing its robustness in the presence of noise.

### E. Position-Wise Cosine Similarity

As illustrated in the proposed TraHGR architecture in Fig. 1, each patch in the in TNet only consists of the temporal information of one sensor for the length of window size (e.g., 200ms, 150ms). As a result, the positional embeddings represent their associated sensors. Therefore, as shown in Fig. 4, the position-wise cosine similarity of the positional embedding vectors in the TNet captures the mutual correlation/entanglement of the sensors in the hand movements. As depicted in Fig. 4, the sensory information is highly correlated for the TraHGR-Base as the smallest network, when the network gets larger (left to right) and the sequence length gets longer (down to up), the network’s capacity to cherry-pick the sensors to associate is increased.

On the other hand, each patch in FNet consists of both temporal and spatial information. As illustrated in Fig. 1, unlike the patching mechanism in TNet, there is temporal information flow from one path to another in the FNet patching mechanism which makes the order of the sequence of patches/information important. These sequential correlations of the patches are expected to be deduced by the FNet. The optimal similarity should result in a matrix with bright colors on the main diagonal and its neighbors. In a sense that consecutive positions are required to be more similar/brighter to reflect the importance of the sequence of patches’ order. As shown in Fig. 5, TraHGR-Huge captures the position meanings better than TraHGR-Large, and TraHGR-Large better than TraHGR-Base for both window sizes 200ms and 150ms. As a consequence, it is possible to conclude that a more complex architecture can improve position embedding inference and includes more location data for transformer encoders. Moreover, as shown

---

| Method                  | Database            | Step size | Window size 200ms | Window size 150ms | Window size 100ms |
|-------------------------|---------------------|-----------|-------------------|-------------------|-------------------|
| CNN [5]                 | DB2 (49 gestures)   | 100ms     | 83.70             | 82.70             | 81.10             |
| Attention-based Hybrid  | CNN-RNN [6]         | 100ms     | 82.20             | -                 | -                 |
| CNN [33]                | DB2 (49 gestures)   | 100ms     | 78.71             | -                 | -                 |
| CNN [32]                | DB2 (49 gestures)   | 100ms     | 77.44             | -                 | -                 |
| CNN [23]                | DB2 (49 gestures)   | 100ms     | 75.27             | -                 | -                 |
| SVM [32]                | DB2 (49 gestures)   | 100ms     | 72.25             | -                 | -                 |
| RF [14]                 | DB2 (49 gestures)   | 100ms     | **86.18**         | 85.43             | 84.13             |
| RF [17]                 | DB2 (49 gestures)   | 100ms     | **86.00**         | 85.22             | 83.95             |
| TraHGR-Huge             | DB2 (49 gestures)   | 100ms     | **86.18**         | 85.43             | 84.13             |
| HDCAM [59]              | DB2-B (17 gestures) | 10ms      | 81.10             | 80.53             | -                 |
| TC-HGR [60]             | DB2-B (17 gestures) | 10ms      | 80.72             | -                 | -                 |
| CNN [32]                | DB2-B (17 gestures) | 100ms     | 82.22             | -                 | -                 |
| CNN [33]                | DB2-B (17 gestures) | 10ms      | -                 | -                 | 83.79             |
| SVM [32]                | DB2-B (17 gestures) | 100ms     | 81.07             | -                 | -                 |
| TraHGR-Huge             | DB2-B (17 gestures) | 100ms     | **88.91**         | 88.14             | 86.46             |
| TraHGR-Huge             | DB2-B (17 gestures) | 100ms     | **88.72**         | 87.97             | 86.25             |
| CNN [32]                | DB2-C (23 gestures) | 100ms     | 72.62             | -                 | -                 |
| SVM [32]                | DB2-C (23 gestures) | 100ms     | 71.08             | -                 | -                 |
| TraHGR-Huge             | DB2-C (23 gestures) | 100ms     | **81.44**         | 79.99             | 78.72             |
| TraHGR-Huge             | DB2-C (23 gestures) | 100ms     | **81.27**         | 79.78             | 78.48             |
| CNN [32]                | DB2-D (9 gestures)  | 100ms     | 89.54             | -                 | -                 |
| SVM [32]                | DB2-D (9 gestures)  | 100ms     | 88.56             | -                 | -                 |
| TraHGR-Huge             | DB2-D (9 gestures)  | 100ms     | **93.84**         | **93.58**         | **91.62**         |
| TraHGR-Huge             | DB2-D (9 gestures)  | 100ms     | **93.74**         | **93.33**         | **91.49**         |
Fig. 4. Position embedding similarities for TNet path in TraHGR-Base, TraHGR-Large, and TraHGR-Huge architectures: (a) window size is 200ms, and (b) window size is 150ms. Each row in each figure represents the cosine similarity between one embedding position and all the other embeddings. The brightness of the pixels in the figures indicates more similarity.

Fig. 5. Position embedding similarities for FNet path in TraHGR-Base, TraHGR-Large, and TraHGR-Huge architectures: (a) window size is 200ms, and (b) window size is 150ms. Each row in each figure represents the cosine similarity between one embedding position and all the other embeddings. The brightness of the pixels in the figures indicates more similarity.

in Fig. 5, for longer window sizes, the sequential nature of sEMG signals can be better encoded. For instance, as shown in Fig. 5(b), the position embeddings for the TraHGR-Base architecture did not adequately infer the concept of positions. As a result, it is reasonable to deduce that the window size has a direct influence on the transformer encoder’s ability to infer position information.

F. Comparison With State-of-the-Art DNN Approaches

TABLE VI provides a comparison between our proposed approach, TraHGR-Huge, and the available methodologies, which shows the superiority of the TraHGR architecture over the experimental results obtained from the state-of-the-art researches [5], [6], [14], [23], [32], [33], [59], [60]. This comparison was evaluated based on the same settings for the DB2 (49 gestures) dataset and its sub-exercises, i.e., DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures). For instance, most of the recent studies, including the Ninapro dataset [14], predominantly employed a window size of 200ms with a step size of 100ms. Taking this into account, besides the reported results for the step size of 10ms in TABLE VI, we conducted additional experiments to evaluate the performance of the proposed model using a step size of 100ms. As demonstrated in TABLE VI, increasing the step size from 10ms to 100ms resulted in minimal performance degradation of the TraHGR, while the proposed approach still outperforms other counterparts even with the larger step size.

In this study, we segmented sEMG signals with three window sizes, i.e., 200ms, 150ms, and 100ms. As shown in TABLE VI, our proposed approach TraHGR-Huge achieved higher accuracy than the existing methodologies when evaluated based on DB2 (49 gestures), DB2-B (17 gestures), DB2-C (23 gestures), DB2-D (9 gestures), different time window sizes, and step size. We compared the proposed architecture with both advanced DNNs and classical ML approaches.

For example, Reference [23] showed the average classification accuracy obtained using all the classical methods such as SVM, RF, K-Nearest Neighbors (K-NN), and LDA on the DB2 (49 gestures) dataset is 60.28%. They achieved the highest gesture recognition accuracy for RF which is 75.27%. Moreover, in Reference [32], they achieved the recognition accuracy 77.44% using SVM over all the movements. In addition, the recognition accuracy of 72.25% is reported in Reference [17] for the RF classifier. For DNN architectures, on the other hand, the best detection accuracy was reported in Reference [5] using CNN, which is 83.70%. As shown in TABLE VI, for window size of 200ms, our proposed architecture achieved 86.18% classification accuracy which is 2.48% higher than the state-of-the-art DNN approach and 8.74% higher than state-of-the-art classical ML method. Moreover, it can be observed that for other window sizes, the classification accuracy of our proposed approach achieved better gesture recognition performances than its counterparts. For example, when the window size is set to 100ms, our proposed approach TraHGR-Huge was able to achieve gesture recognition accuracy of 84.13%, but using the proposed approach of [5], accuracy of 81.1% is achieved. It should be noted that the accuracy of 84.13% obtained by TraHGR-Huge with a window size of 100ms is still higher than the case where the window size in Reference [5] has doubled, i.e., 200ms. We also evaluated and compared our proposed method for DB2-B (17 gestures), DB2-C (23 gestures), and
TABLE VII

The Average Accuracy of Hand Gesture Recognition Across All Subjects in the Second Experiment of Ninapro DB5 Dataset on the Window Size of 260ms. The Average Accuracy Is Reported on 5 and 6 Repetitions for All Models in Ninapro DB5 Dataset

| Repetitions (Rep.) Used for Training/Fine-tuning | ConvNet [53] | ConvNet+TL [53] | TraHGR-Huge | TraHGR-Huge+TL |
|-----------------------------------------------|--------------|----------------|-------------|----------------|
| Rep. 1, 2, 3, 4                               | 66.30 ± 3.77 | 68.98 ± 4.09   | 71.21 ± 1.99 | 74.63 ± 2.52   |
| Rep. 1, 2, 3                                  | 61.91 ± 3.94 | 65.16 ± 4.46   | 68.62 ± 2.07 | 69.01 ± 2.77   |
| Rep. 1, 2                                     | 55.65 ± 4.38 | 60.12 ± 4.79   | 58.68 ± 2.81 | 62.07 ± 2.70   |
| Rep. 1                                        | 46.06 ± 6.09 | 49.41 ± 5.82   | 51.33 ± 2.93 | 53.42 ± 3.31   |

DB2-D (9 gestures) with the previous studies [32], [33], which demonstrates the superiority of our hybrid transformer-based framework.

Authors in [35] introduced a hybrid CNN-LSTM model achieving 79% average accuracy on the window size of 200ms as shown in TABLE VI. They reduced the number of parameters in the proposed model using dilated LSTM, resulting in 1,102,801 parameters for 17 gesture classifications. However, as shown in TABLE VI, TraHGR-Huge outperforms [35] for 17 gesture classification with less number of parameters (832, 659).

G. Transfer Learning Impact on TraHGR Performance

In this experiment, The 5th Ninapro database [17], referred to as the DB5, is used for the ease of comparison with Ref. [53]. The DB5 dataset is recorded with two Thalmic Myo-armbands recording muscular activity at a rate of 200Hz. The DB5 dataset, in particular, consists of signals collected from 10 users executing 52 actions/movements. Each movement in the DB5 dataset is repeated 6 times, each lasting for 5 seconds followed by 3 seconds of rest. The DB5 dataset is provided in three sets of exercises [17]. In this paper, we only consider data collected by the lower armband in DB5 in the second exercise of the DB5 to follow the same criteria in [53] and also have a fair comparison. Moreover, out of 6 movement repetition for each target user, following [53], the first four repetitions are used to fine-tune the pre-trained network, and the last two repetitions serve as the test set.

TABLE VII shows the average accuracy on the second experiment of the Ninapro DB5 dataset. As shown in TABLE VII, the TraHGR-Huge outperforms ConvNet [53] whether the training process of the network is involved with the TL stage or solely trained for each user. In TABLE VII, the networks without TL training stage are independently trained for each user. However, to integrate TL techniques into the training process, we conducted a typical TL method to utilize the knowledge learned in the source domain to promote the learning process in a target domain. Specifically, given a user as the target, in the first stage, the training sets of the remaining nine participants/users are employed to pre-train the TraHGR-Huge network. Then, to fine-tune the pre-trained network, the weights of the TNet and FNet in TraHGR-Huge are maintained intact by freezing them, and the non-frozen parts of the network are updated using one, two, three, or four repetitions of the target data (see TABLE VII). As shown in TABLE VII, using transfer learning as a domain adaptation approach is conducted to performance improvement of both the TraHGR-Huge and ConvNet models compared to their corresponding user-specific trained models. When comparing our transformer-based model to ConvNet with convolutional structure, we can infer that TraHGR-Huge achieves higher accuracies, demonstrating the proposed model’s ability to extract more useful representations from raw sEMG data.

H. Ablation Study

For the proposed hybrid architectures such as TraHGR-Huge, TraHGR-Large, and TraHGR-Base, the classification accuracy is calculated using the prediction values y obtained from Eq. (10). To show that our proposed architecture based on a developed hybrid strategy has great potentials for improving gesture recognition accuracy, we also calculated the other two accuracies, i.e., yTNet or yFNet, based on the Eq. (9). More specifically, we trained the hybrid architectures by computing the loss function in Eq. (11). However, output y is used to calculate the accuracy, reported in this paper. Here, in Fig. 6, it is shown that the accuracy obtained using the y is better than those calculated using the yTNet or yFNet for DB2 (49 gestures) and its sub-exercises. In particular, from Fig. 6, it can be observed that the hybrid architecture takes advantage of two parallel paths and improved the recognition accuracy.
The proposed transformer-based hybrid architecture to other machine learning fields.

REFERENCES

[1] N. Jiang, S. Dosen, K. R. Müller, and D. Farina, “Myoelectric control of artificial limbs—Is there a need to change focus?” IEEE Signal Process. Mag., vol. 29, no. 5, pp. 150–152, Sep. 2012.

[2] D. Farina, R. Merletti, and R. M. Enoka, “The extraction of neural strategies from the surface EMG,” J. Appl. Physiol., vol. 96, pp. 1486–1495, Jan. 2004.

[3] D. Farina et al., “The extraction of neural information from the surface EMG for the control of upper-limb prostheses: Emerging avenues and challenges,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 22, no. 4, pp. 797–809, Jul. 2014.

[4] M. Ergeneci, K. Gokcesu, E. Ertan, and P. Kosmas, “An embedded, eight channel, noise canceling, wireless, wearable sEMG data acquisition system with adaptive muscle contraction detection,” IEEE Trans. Biomed. Circuits Syst., vol. 12, no. 1, pp. 68–79, Feb. 2018.

[5] W. Wei, Q. Dai, Y. Du, M. Kankanhalli, and W. Geng, “Surface-electromyography-based gesture recognition by multi-view deep learning,” IEEE Trans. Biomed. Eng., vol. 66, no. 10, pp. 2964–2973, Oct. 2019.

[6] Y. Hu, Y. Wong, W. Wei, Y. Du, M. Kankanhalli, and W. Geng, “A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition,” PLoS ONE, vol. 13, no. 1, e0196649.

[7] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, “Gesture recognition by instantaneous surface EMG images,” Sci. Rep., vol. 6, no. 1, p. 36571, Nov. 2016.

[8] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, “FS-HGR: Few-shot learning for hand gesture recognition via electromyography,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 29, no. 10, pp. 1004–1015, 2021.

[9] 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 myococontrol,” IEEE Trans. Neural Sys. Rehabil. Eng., vol. 23, no. 2, pp. 189–198, Mar. 2015.

[10] A. Vaswani et al., “Attention is all you need,” 2017, arXiv:1706.03762.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.

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

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

[14] M. Atzori et al., “Electromyography data for non-invasive naturally-controlled robotic hand prostheses,” Sci. Data, vol. 1, no. 1, pp. 1–13, 2014.

[15] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo, “Movement error rate for evaluation of machine learning methods for sEMG-based hand movement classification,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 22, no. 4, pp. 735–744, Jul. 2014.

[16] M. Atzori et al., “Characterization of a benchmark database for myoelectric movement classification,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 23, no. 1, pp. 73–83, Jan. 2015.

[17] S. Pizzolato, L. Tagliapietra, M. Reggiani, M. Fongiolo, M. Reggiani, H. Müller, and M. Atzori, “Comparison of six electromyography acquisition setups on hand movement classification tasks,” PLoS ONE, vol. 12, no. 10, pp. 1–7, 2017.

[18] A. Ameri, M. A. Akhæe, E. Scheme, and K. Englehart, “Regression convolutional neural network for improved simultaneous EMG control,” J. Neural Eng., vol. 16, no. 3, Jun. 2019, Art. no. 036015.

[19] C. Li, J. Ren, H. Huang, B. Wang, Y. Zhu, and H. Hu, “PCA and deep learning based myoelectric grasping control of a prosthetic hand,” Biomed. Eng. OnLine, vol. 17, no. 1, p. 107, Dec. 2018.

[20] D. Espósito et al., “A piezoresistive array armband with reduced number of sensors for hand gesture recognition,” Frontiers Neurorobot., vol. 13, p. 114, Jan. 2020.

[21] M. Tavakoli, C. Benussi, P. Alhais Lopes, L. B. Osorio, and A. T. de Almeida, “Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier,” Biomed. Signal Process. Control, vol. 46, pp. 121–130, Sep. 2018.

[22] G. R. Naik, A. H. Al-Timemy, and H. T. Nguyen, “Transradial amputee gesture classification using an optimal number of sEMG sensors: An approach using ICA clustering,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 24, no. 8, pp. 837–846, Aug. 2016.
[23] M. Atzori, M. Cognolato, and H. Müller, “Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands,” Frontiers Neurorobot., vol. 10, p. 9, Sep. 2016.

[24] A. K. Clarke et al., “Deep learning for robust decomposition of high-density surface EMG signals,” IEEE Trans. Biomed. Eng., vol. 68, no. 2, pp. 526–534, Feb. 2021.

[25] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, “Surface EMG-based hand gesture recognition via hybrid and dilated deep neural network architectures for neurorobotic prostheses,” J. Med. Robot. Res., vol. 5, pp. 1–12, Mar. 2020.

[26] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, “XceptionTime: Independent time-window XceptionTime architecture for hand gesture classification,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 1304–1308.

[27] E. Rahimian, S. Zabihi, A. Asif, S. F. Atashzar, and A. Mohammadi, “Few-shot learning for decoding surface electromyography for hand gesture recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 1300–1304.

[28] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, “Semg-based hand gesture recognition via dilated convolutional neural networks,” in Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP), Nov. 2019, pp. 1–5.

[29] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, “A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface,” Pattern Recognit. Lett., vol. 119, pp. 131–138, Mar. 2019.

[30] U. Cotè-Allard, E. Campbell, A. Phinyomark, F. Laviolette, B. Gosselin, and E. Scheme, “Interpreting deep learning features for myoelectric control: A comparison with handcrafted features,” Frontiers Bioeng. Biotechnol., vol. 8, p. 158, Mar. 2020.

[31] R. N. Khushaba and S. Kodagoda, “Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control,” in Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV), Dec. 2012, pp. 1534–1539.

[32] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, “Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network,” Frontiers Neurosci., vol. 11, p. 379, Jul. 2017.

[33] Z. Ding, C. Yang, Z. Tian, C. Yi, Y. Fu, and F. Jiang, “SEMGB-based gesture recognition with convolution neural networks,” Sustainability, vol. 10, no. 6, p. 1865, Jun. 2018.

[34] F. Quivira, T. Koike-Akino, Y. Wang, and D. Erdogmus, “Translating sEMG signals to continuous hand poses using recurrent neural networks,” in Proc. IEEE EMBS Int. Conf. Biomed. Health Informat. (BHI), Mar. 2018, pp. 166–169.

[35] T. Sun, Q. Hu, P. Gulati, and S. F. Atashzar, “Temporal dilation of deep LSTM for agile decoding of sEMG: Application in prediction of upper-limb motor intention in NeuroRobotics,” IEEE Robot. Autom. Lett., vol. 6, no. 4, pp. 6212–6219, Oct. 2021.

[36] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, arXiv:1803.01271.

[37] J. Guan, W. Wang, P. Feng, X. Wang, and W. Wang, “Low-dimensional denoising embedding transformer for ECG classification,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 1285–1289.

[38] Y. Song, X. Jia, L. Yang, and L. Xie, “Transformer-based spatial–temporal feature learning for EEG decoding,” 2021, arXiv:2106.11170.

[39] T. B. Brown et al., “Language models are few-shot learners,” 2020, arXiv:2005.14165.

[40] Y. Wang et al., “Transformer-based acoustic modeling for hybrid speech recognition,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 6874–6878.

[41] T. Bao, S. Q. Xie, P. Yang, P. Zhou, and Z.-Q. Zhang, “Toward robust, adaptive real upper-limb motion estimation using machine learning and deep learning—a survey in myoelectric control,” IEEE J. Biomed. Health Informat., vol. 26, no. 8, pp. 3822–3835, Aug. 2022.

[42] M. Kim, W. K. Chung, and K. Kim, “Subject-independent sEMG pattern recognition by using a muscle source activation model,” IEEE Robot. Autom. Lett., vol. 5, no. 4, pp. 5175–5180, Oct. 2020.

[43] K. Watanabe, M. Kouzaki, M. Ogawa, H. Akima, and T. Moritani, “Relationships between muscle strength and multi-channel surface EMG parameters in eighty-eight elderly,” Eur. Rev. Aging Phys. Activity, vol. 15, no. 1, pp. 1–15, Dec. 2018.

[44] E. C. Hill et al., “Effect of sex on torque, recovery, EMG, and MMG responses to fatigue,” J. Musculoskelet. Neuronal Interact., vol. 16, no. 4, pp. 310–317, 2016.

[45] J. He, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, “Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns,” J. NeuroEng. Rehabil., vol. 12, no. 1, pp. 1–10, Dec. 2015.

[46] M. Jochumsen, A. Waris, and E. N. Kamavuako, “The effect of arm position on classification of hand gestures with intramuscular EMG,” Biomed. Signal Process. Control, vol. 43, pp. 1–8, Jan. 2018.

[47] K.-T. Kim, C. Guan, and S.-W. Lee, “A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 1, pp. 94–103, Jan. 2020.

[48] A. Ameri, M. A. Akhare, E. Scheme, and K. Englehart, “A deep transfer learning approach to reducing the effect of electrode shift in EMG pattern recognition-based control,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 2, pp. 370–379, Feb. 2020.

[49] Z. Yu, J. Zhao, Y. Wang, L. He, and S. Wang, “Surface EMG-based instantaneous hand gesture recognition using convolutional neural network with the transfer learning method,” Sensors, vol. 21, no. 7, p. 2540, Apr. 2021.

[50] F. Demir, V. Bajic, M. C. Ince, S. Taran, and A. Sengr, “Surface EMG signals and deep transfer learning-based physical action classification,” Neurocomput., vol. 31, no. 12, pp. 8455–8462, 2019.

[51] J. J. Bird, J. Kobylarz, D. R. Faria, A. Ekárt, and E. P. Ribeiro, “Cross-domain MLP and CNN transfer learning for biological signal processing: EEG and EMG,” IEEE Access, vol. 8, pp. 54789–54801, 2020.

[52] U. Cotè-Allard et al., “Deep learning for electromyographic hand gesture signal classification using transfer learning,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 27, no. 4, pp. 760–771, Apr. 2019.

[53] M. Asfour, C. Menon, and X. Jiang, “Feature-classifier pairing compatibility for sEMG signals in hand gesture recognition under joint effects of processing procedures,” Bioengineering, vol. 9, no. 11, pp. 634, Nov. 2022.

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

[55] T. R. Farrell and R. F. Weir, “The optimal controller delay for myoelectric prostheses,” IEEE Trans. Neural Syst. Rehabil. Eng., vol. 15, no. 1, pp. 111–118, Mar. 2007.

[56] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[57] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[58] S. Zabihi, E. Rahimian, A. Asif, and A. Mohammadi, “Light-weight CNN-attention based architecture for hand gesture recognition via electromyography,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2023, pp. 1–5.

[59] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, “Hand gesture recognition using temporal convolutions and attention mechanism,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2022, pp. 1196–1200.

[60] F. Wilcoxon, “Individual comparisons by ranking methods,” Biometrics Bull., vol. 1, no. 6, pp. 80–83, 1945.