Multimodal Surface EMG Hand Gesture Recognition Using Query-Based Transformers for Prosthetic Control

Federico Del Pup[a,b,*], Elisa Tentori[b,c] and Manfredo Atzori[b,d,e]

[a]Department of Information Engineering, University of Padua, Padua, 35131, Italy
[b]Padova Neuroscience Center, University of Padua, Padua, 35129, Italy
[c]Department of Biomedical Sciences, University of Padua, Padua, 35131, Italy
[d]Department of Neuroscience, University of Padua, Padua, 35121, Italy
[e]Information Systems Institute, University of Applied Sciences Western Switzerland (HES-SO Valais), Sierre, 3960, Switzerland

ARTICLE INFO

Keywords:
Cross-attention
Deep learning
Electromyography
Hand gesture recognition
Machine learning
Multimodal
Transformer

ABSTRACT

Hand gesture recognition via surface electromyography (sEMG) is fundamental to human-machine interaction and prosthetic control. In this field, deep learning approaches have become the gold standard. However, current architectures struggle to scale; model performance typically decreases as the number of hand movements increases. Performance degradation is tied to the increased statistical complexity of decoding expanded gesture sets and compounded by the limitations of state-of-the-art approaches, which primarily rely on low-latency unimodal convolutional architectures. Convolutions operate locally, limiting model's ability to capture long-range sequential patterns. Unimodal setups cannot leverage complementary information from coordinated signals characterizing movement execution, such as inertial and eye-tracking data. These limitations motivate architectures that integrate local and global features across multimodal physiological sequences. To bridge this gap, this study introduces EMG-CrossFormer, an end-to-end hybrid convolutional-transformer for seamless multimodal integration. EMG-CrossFormer combines representations from an arbitrary number of unimodal encoders through cascaded cross-attention fusion layers, and decodes the fused representations using learnable gesture queries. EMG-CrossFormer was evaluated on four NinaPro datasets (DB2, DB3, DB7, and DB10) and benchmarked against six state-of-the-art models using increasing number of modalities. Using only sEMG signals, EMG-CrossFormer achieved mean accuracies of 72.33%, 52.48%, 79.16%, and 73.49% on DB2, DB3, DB7, and DB10, respectively, consistently achieving the highest performance. Incorporating inertial signals improved performance to 90.66%, 80.40%, 92.79%, and 92.06%. These results show that joint local-global feature modeling improves sEMG-only decoding and that multimodal fusion substantially amplifies this benefit, underscoring the value of both design principles for complex hand gesture recognition in prosthetics.

1. Introduction

Surface electromyography (sEMG)-controlled upper-limb prostheses have been used clinically since the 1960s [1], and decades of subsequent research have progressively refined their control algorithms and improved their reliability. These myoelectric devices have been shown to enhance the quality of life for trans-radial amputees by partially restoring limb motor functions [2, 3, 4].

To date, commercial myoelectric solutions integrate pattern-recognition algorithms for automatic sEMG gesture recognition, typically supporting a limited set of functional grasps with high accuracy and minimal latency [5]. Concurrently, advances in the computational capabilities of embedded systems have driven a shift in the research community from classical machine learning approaches to deep learning paradigms [6]. However, this transition has not yet been reflected in commercial pattern-recognition systems, which still predominantly rely on classical algorithms.

In research settings, several studies have reported outstanding decoding performance, frequently exceeding 90% accuracy on commonly used gesture sets comprising up to 17 hand movements representative of activities of daily living [7, 8, 9, 10]. However, current literature in this domain often suffers from a lack of reproducibility, complicating the direct benchmarking of different architectures under identical experimental settings. Furthermore, common evaluation protocols frequently omit various movements identified as highly similar by quantitative taxonomies [11], simplifying the classification task, potentially inflating reported performance, and decreasing the usefulness of benchmark datasets. These evaluation gaps make it difficult to critically assess the true benefits of deep learning frameworks over classical machine learning models, such as Random Forests (RF) or Support Vector Machines (SVM), which continue to offer highly competitive performance [12].

These limitations of current deep learning approaches become particularly evident when models are scaled to larger sets of hand gesture. Under these demanding scenarios, the performance of deep learning models drops significantly, often falling below that of machine learning classifiers. For example, in [13], the authors evaluated a shallow

*Corresponding author.
This document is the results of the research project by the European Union's Horizon Europe research and innovation programme under Grant agreement no 101137074 - HEREDITARY.
federico.delpup@unipd.it (F. Del Pup);
elisa.tentori@unipd.it (E. Tentori);
manfredo.atzori@unipd.it (M. Atzori)
ORCID(s): 0009-0004-0698-962X (F. Del Pup);
0000-0002-4755-1577 (E. Tentori); 0000-0001-5397-2063 (M. Atzori)

Convolutional Neural Network (CNN) against a set of 51 distinct hand movements from the Non Invasive Adaptive Prosthetics (NinaPro) dataset [14], revealing a performance degradation of more than 10% compared to a RF across both intact and amputee cohorts.

The reported reduction in performance highlights some limitations of the existing deep learning frameworks. State-of-the-art networks rely predominantly on unimodal convolutional architectures optimized for low latency. While conventional convolutions enable highly parallelized and efficient computations that keep system latency below the recommended real-time threshold of 125 ms [15], they are inherently limited to extracting local features. This local receptive field limits the model's ability to distinguish complex neural activation patterns from noise, thereby reducing decoding accuracy and preventing the model from fully exploiting the performance gains offered by deep neural networks. In contrast, attention layers offer a complementary solution that allows sEMG deep learning models to capture longer temporal dynamics [16]. However, their integration into low-latency, resource-constrained models remains scarcely investigated.

Similarly, unimodal (sEMG-only) approaches prevent these architectures from effectively leveraging complementary information from other signal modalities, such as accelerometer data, which can help characterize both motor intent and the physical execution of movement. Previous work has demonstrated that combining sEMG signals with accelerometer data can improve classification accuracy in amputee subjects to levels comparable to intact individuals [17, 18]. Consistent with these findings, the authors in [19] achieved comparable results with a novel multimodal convolutional model that integrates sEMG and accelerometer data within a compact, end-to-end framework.

Despite the benefits highlighted, multimodal solutions integrating further modalities (e.g., eye tracking, scene videos) remain less investigated. These limitations motivate the development of novel architectures capable of simultaneously extracting and combining local and global features across heterogeneous multimodal physiological sequences.

Contributions: To bridge this gap, this study introduces EMG-CrossFormer, an end-to-end, hybrid convolutional-transformer framework designed for seamless multimodal signal integration. EMG-CrossFormer combines representations from an arbitrary number of unimodal encoders through cascaded cross-attention fusion layers. Then, it decodes the fused representations via cross-attention with learnable gesture queries, inspired by consolidated query-based decoding approaches applied in other domains (e.g., DETR [20]). EMG-CrossFormer was evaluated on four NinaPro databases (DB2, DB3, DB7, and DB10) and benchmarked against six state-of-the-art models using an incremental number of input modalities. While this study focuses on the integration of sEMG with accelerometer and eye-tracking data, EMG-CrossFormer has been designed to be easily extended to additional data modalities, facilitating the development of multimodal architectures incorporating, for instance, scene camera or EEG data.

Paper structure: the outline of this paper is as follows. Section 2 describes in detail the model architecture and the experimental setting. Section 3 presents the comparative analysis between EMG-CrossFormer and the other selected models using an incremental number of data modalities. Section 4 critically discusses the results, highlighting potential limitations and future directions. Finally, a conclusion is drawn in section 5.

2. Methods

This section outlines the methodological design of the study, providing thorough details on the procedure and the architecture, and fostering the reproducibility of the results. Specifically, it covers dataset selection, data preprocessing, model architecture, training hyperparameters, performance evaluation, and statistical analysis.

2.1. Datasets

The proposed EMG-CrossFormer model was evaluated using four distinct datasets from the public NinaPro repository[1] [12]: datasets 2, 3, 7, and 10, referred to as DB2, DB3, DB7, and DB10, respectively. Table 1 summarizes their main acquisition configurations.

The datasets were selected to ensure a comprehensive evaluation of decoding performance across heterogeneous cohorts of both intact and trans-radial amputee subjects, while also investigating increasing levels of sensor multimodality. DB2, DB3, and DB7 provide synchronized surface electromyography (sEMG) and tri-axial accelerometer recordings acquired with a 12-channel Delsys Trigno™ system (Delsys, Natick, MA, USA). They include a broad set of hand-motion and grasping movements collected from both intact subjects (DB2, DB7) and trans-radial amputee subjects (DB3, DB7). Movements were organized into three sessions, referred to as Exercises B, C, and D, comprising 17, 23, and 9 movements, respectively. Each gesture was repeated six times; active trials lasted 5 seconds and were followed by 3 seconds of rest. DB7 only includes Exercises B and C. Therefore, Exercise D was excluded from DB2 and DB3 to harmonize the label space across the three datasets, yielding 40 discrete gestures plus the resting state. sEMG signals were acquired at a sampling rate of 2 kHz, whereas accelerometer signals were sampled at 148 Hz and upsampled by the original authors to match the sEMG sampling rate. All subjects were included in the analysis except for two DB3 subjects, for whom the number of electrodes was reduced due to insufficient residual limb space, according to the NinaPro authors' usage notes [12].

DB10 includes 10 grasping movements and the resting state, selected from exercises B and C based on activities of daily living. Data were collected from a cohort of 30 intact subjects and 15 trans-radial amputee subjects [21]. Each movement repetition lasted between 5 and 6 seconds and was

[1] [Online] Available: https://ninapro.hevs.ch/

Table 1
Main acquisition configurations of the NinaPro datasets used in this work.

Dataset Name	Subjects*	Device	Channels	sEMG sampling rate [Hz]	Modalities used	Movements	Repetitions
DB2	40	Delsys Trigno	12	2000	sEMG + ACC	40 + rest	6
DB3**	11(A)	Delsys Trigno	12	2000	sEMG + ACC	40 + rest	6
DB7	20 + 2(A)	Delsys Trigno (Wireless)	12	2000	sEMG + ACC	40 + rest	6
DB10***	30 + 15(A)	Delsys Trigno (Wireless)	12	1926	sEMG + ACC + Gaze	10 + rest	10

*A stands for Amputees; **Subjects 6 and 7 were discarded according to usage notes provided in [12]
***Six subjects and unreliable repetitions were discarded according to usage notes provided in [21]

followed by 4 seconds of rest. DB10 was acquired with three recording modalities at the same time: sEMG, accelerometer and gaze signals. sEMG signals were recorded using a 12-channel Delsys Trigno™ Wireless system at a sampling frequency of 1926 Hz. Upper-limb kinematics were captured via tri-axial accelerometers at a native sampling rate of 148 Hz, and eye-gaze dynamics were recorded using Tobii Pro Glasses 2 at 100 Hz. Both accelerometer and gaze signals were upsampled by the original authors to match the sEMG sampling rate and ensure temporal alignment across modalities [21]. Following the authors' usage notes, six subjects were discarded due to poor signal quality, and repetitions marked as unreliable by the original authors were excluded [21].

2.2. Data Preprocessing

Raw sEMG signals were preprocessed using the following pipeline:

- *scale conversion*: sEMG signals were converted to millivolts to improve numerical stability and avoid representation issues during mixed-precision (FP16) training.
- *DC component removal*: the channel mean was subtracted from each sEMG channel, which is equivalent to removing the DC component.
- *Filtering*: sEMG signals were filtered using a second-order (12 dB/oct) forward-backward Butterworth band-pass filter with cutoff frequencies of 20 Hz and 500 Hz. The filter order and cutoff frequencies were selected according to guidelines for biomechanical and clinical applications [22]. Power-line noise and its harmonics had already been removed by the original authors using a Hampel filter [23].
- *Resampling*: sEMG signals were resampled to 1 kHz to reduce memory footprint and the number of floating-point operations (FLOPs) required by the deep learning architectures.
- *Window extraction*: sEMG data were partitioned into windows of 100, 150, or 200 ms with a 10% shift (corresponding to 90% overlap) to increase the number of samples.
- *Rest class balancing*: windows belonging to the resting class were undersampled to match the gesture class ratios.

Accelerometer data underwent only resampling and window extraction to preserve temporal consistency. Gaze signals, represented as (x, y) coordinates in image space, were processed in the same way after short intervals of missing values were imputed by linear interpolation, following the guidelines provided in [24].

2.3. EMG-CrossFormer Architecture

EMG-CrossFormer is a hybrid convolutional-transformer model designed to jointly process and fuse multiple input modalities (signals in this study) for hand-movement decoding. As illustrated in Figure 1, the architecture consists of four main modules: modality-specific backbones, which map each input modality to a compact sequence of feature tokens; a fusion module that combines the token sequences through a cascade of cross-attention fusion blocks; a transformer decoder with learnable gesture queries; and a two-layer feed-forward network (FFN) for hand-gesture recognition.

The model is highly flexible and was designed to be easily adaptable to different experimental setups, including different types and numbers of input modalities. Unimodal backbones may differ across modalities, provided that they output representations in the form of a sequence of tokens. Furthermore, the number of modalities can also vary by adding or removing fusion blocks in the cascade.

A PyTorch [25] implementation of the model is provided in the openly available source code[2]. Initialization hyperparameters are also reported in the Supplementary Materials.

2.3.1. Unimodal Backbone

Let m denote the input modality, C_m the number of channels and W_m the number of samples within the input window. Given an input multi-channel signal $\mathbf{x}_{\text{sig}_m} \in \mathbb{R}^{C_m \times W_m}$, the m-th unimodal backbone generates a compact representation $\mathbf{z}_{\text{enc}_m} \in \mathbb{R}^{L_m \times d}$ (Figure 1), where L_m is the sequence length (number of output tokens) and d is the token embedding dimension. L_m may vary across modalities, particularly when the corresponding signals have different sampling rates. By contrast, d is kept fixed across modalities to enable the cross-attention in the fusion and decoding modules.

To assess the architecture's adaptability to different encoder designs, two unimodal backbone implementations were evaluated: a 1D depthwise convolutional encoder and a 2D multi-scale convolutional encoder.

[2][Online] Available: https://github.com/deepPNClab/emg-crossformer

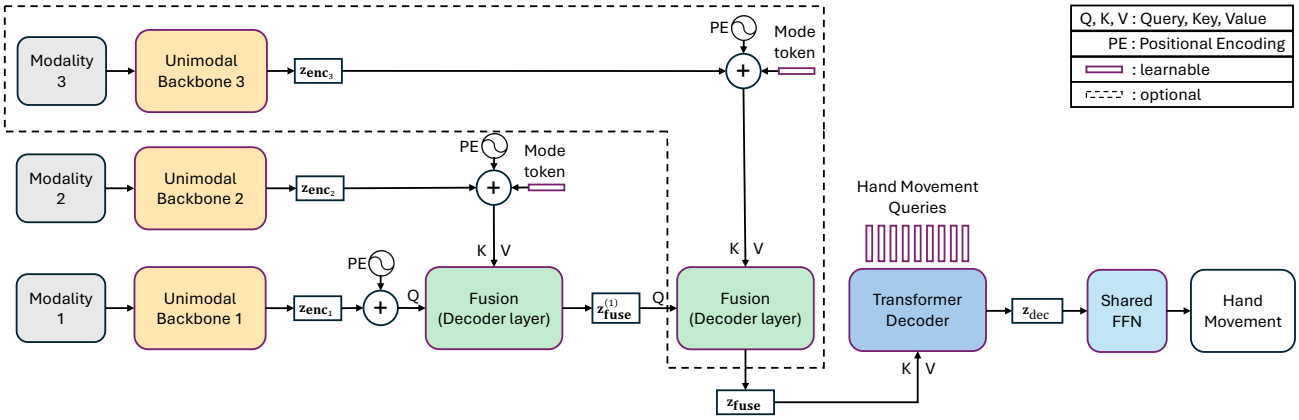

Figure 1: Schematic representation of EMG-CrossFormer. The model combines representations ($\mathbf{z}_{\text{enc}_m}$) from unimodal backbones through cascaded fusion layers (transformer decoder layers). The transformer decoder block decodes the fused representation ($\mathbf{z}_{\text{fuse}}$) using learnable hand movement queries. A final shared FFN outputs the hand movement predictions from the decoded representation ($\mathbf{z}_{\text{dec}}$). EMG-CrossFormer can integrate an arbitrary number of modalities. To illustrate this extensibility, the third modality branch and its corresponding fusion layer are shown as optional (dashed box).

The 1D depthwise implementation, based on [26], extracts channel-specific temporal features through stacked depthwise convolutions and average pooling, with low parameter cost. Each block doubles the feature dimension while halving the temporal resolution. Because convolutions are depthwise, this implementation maintains a low parameter count and restricts feature extraction to temporal patterns, leaving the modeling of inter-channel patterns to subsequent stages of the architecture.

The 2D multi-scale implementation, based on [7], treats the multi-channel signals as a single-channel pseudo-image and applies parallel convolutional branches with different kernel sizes, to capture patterns at multiple frequency scales. The extracted features are subsequently combined through separable convolutions to generate a compact representation while reducing the parameter count. Compared to the 1D depthwise backbone, the 2D multi-scale architecture extracts spatio-temporal features at the cost of increased parameters and computational load.

To integrate these backbones into the EMG-CrossFormer architecture while preserving the architectural designs proposed by the original authors, a learnable linear projection layer is appended to each encoder to map its output into a shared d-dimensional embedding space. The resulting EMG-CrossFormer variants are denoted as EMG-CF_{1D} and EMG-CF_{2D}, respectively.

2.3.2. *Fusion Layers*

The fusion stage combines a variable number M of input modalities through a sequential cascade of non-causal transformer decoder layers. Let $\mathbf{z}_{\text{enc}_m} \in \mathbb{R}^{L_m \times d}$ represent the compact tokenized sequence generated by the m-th unimodal backbone, with $m \in \{1, \dots, M\}$. Here, $m = 1$ denotes the primary sEMG modality, while $m > 1$ denotes subsequent auxiliary signals.

Before fusion, a sinusoidal positional encoding $E^m_{\text{pos}} \in \mathbb{R}^{L_m \times d}$ is injected into each unimodal feature set. For auxiliary modalities, a learnable modality-specific bias, defined as the mode token $\mathbf{t}^m_{\text{mode}} \in \mathbb{R}^{1 \times d}$, is also added directly to the sequence (see "Mode token" in Figure 1). This token acts as an indicator to help fusion blocks contextualize non-sEMG signal representations. For each modality, the processed representation $\bar{\mathbf{z}}_{\text{enc}_m}$ is defined as:

$$\bar{\mathbf{z}}_{\text{enc}_m} = \begin{cases} \mathbf{z}_{\text{enc}_m} + E^m_{\text{pos}} & \text{if } m = 1 \\ \mathbf{z}_{\text{enc}_m} + E^m_{\text{pos}} + \mathbf{1}_{L_m} \mathbf{t}^m_{\text{mode}} & \text{if } m > 1 \end{cases} \tag{1}$$

where $\mathbf{1}_{L_m} = (1, \dots, 1)^T \in \mathbb{R}^{L_m \times 1}$.

The mixing of representations across modalities is performed via a directional cascade, as schematized in Figure 1. For notational reasons, the representations of the first primary modality are considered the initial state of the fusion stage, such that $\mathbf{z}^{(1)}_{\text{fuse}} = \bar{\mathbf{z}}_{\text{enc}_1}$. For each subsequent auxiliary modality $m = 2, \dots, M$, the $(m-1)$-th fusion block applies the set of operations of a non-causal transformer decoder layer.

More formally, the output of the m-th block in the cascade is described as:

$$\mathbf{z}^{(m)}_{\text{fuse}} = \text{LN}\Big(\hat{\mathbf{z}}^{(m)} + \text{FFN}\big(\hat{\mathbf{z}}^{(m)}\big)\Big), \tag{2}$$

where

$$\begin{aligned} \hat{\mathbf{z}}^{(m)} &= \text{LN}\Big(\tilde{\mathbf{z}}^{(m)} + \text{MHA}\big(\tilde{\mathbf{z}}^{(m)}, \bar{\mathbf{z}}_{\text{enc}_m}, \bar{\mathbf{z}}_{\text{enc}_m}\big)\Big) \\ \tilde{\mathbf{z}}^{(m)} &= \text{LN}\Big(\mathbf{z}^{(m-1)}_{\text{fuse}} + \text{MHA}\big(\mathbf{z}^{(m-1)}_{\text{fuse}}, \mathbf{z}^{(m-1)}_{\text{fuse}}, \mathbf{z}^{(m-1)}_{\text{fuse}}\big)\Big) \end{aligned} \tag{3}$$

Here, LN denotes the layer normalization, FFN a pointwise feed-forward network, and MHA the multi-head attention mechanism. For an in-depth description of the multi-head attention mechanism, the reader is referred to the original implementation of the transformer model [27].

Following the final cascade layer, the module outputs a highly descriptive, multimodal representation of the input signals $\mathbf{z}_{\text{fuse}} = \mathbf{z}_{\text{fuse}}^{(M)} \in \mathbb{R}^{L_1 \times d}$. This representation fully compresses the dynamics of all available modalities while preserving the native token sequence length L_1 of the underlying sEMG driver.

In this study, transformer decoder layers were initialized with an embedding dimension $d = 128$, 8 heads, SwiGLU activation [28], and no dropout.

2.3.3. Query-based decoder

The fused multimodal representation $\mathbf{z}_{\text{fuse}} \in \mathbb{R}^{L_1 \times d}$ is fed into a query-based transformer decoder. Following the formulation originally introduced for object detection tasks in computer vision [20], this module implements the standard transformer decoder operations described in Equation 2, using a set of learnable query embeddings as decoder inputs. Analogously, EMG-CrossFormer uses learnable hand gesture queries, each encoding a candidate hand movement. Through self-attention among queries and cross-attention over $\mathbf{z}_{\text{fuse}}$, the model jointly reasons over all gestures, capturing pairwise relationships while using the multimodal fused representation as contextual information. In this study, the number of layers in the query-based decoder was set to 4, with each layer initialized using the same hyperparameters adopted for the fusion stage layers.

2.3.4. Feed-forward network

Final predictions are produced by a shared two-layer Feed-Forward Network (FFN) applied independently to each decoded hand gesture query. The hidden layer has dimensionality $d = 128$ and is followed by a ReLU activation. The FFN outputs class logits for each query, which are subsequently normalized with a softmax function to obtain class probabilities.

2.3.5. Model configuration

In the unimodal setting, only sEMG data are used. To apply the fusion architecture within this setting, the sEMG signals are divided into two model input modalities: forearm channels, corresponding to the first eight channels, and upper-arm channels, corresponding to the last four channels. The forearm channels are provided as the primary input modality, while the upper-arm channels are provided as an auxiliary input modality, potentially allowing the model to learn muscle synergies [29, 30]. In multimodal settings, the model receives sEMG together with accelerometer signals and, when available, gaze information. In this case, sEMG is kept as a single primary input modality, rather than being split into forearm and upper-arm channels. Accelerometer signals define the second input modality, while gaze information is used as the third input modality when present. Additional details regarding model implementation, auxiliary training outputs, and hyperparameter selection are provided in the Supplementary Materials.

2.4. Implementation details

Deep learning models were implemented and trained using PyTorch [25], while conventional machine learning models were implemented with Scikit-learn [31]. Statistical analyses were performed using SciPy [32] and statsmodels [33]. Figures were generated with Seaborn [34]. Experiments were conducted on the Department of Neuroscience computing cluster equipped with four NVIDIA A30 GPUs. Further implementation details are available in the open-source codebase.

2.4.1. Data partition

Results were obtained using an intra-subject evaluation protocol, consistent with the domain [19, 10], where prosthetic devices are required to work robustly on a specific subject. In this setting, models are trained and evaluated on different repetitions of hand movements from the same subject. Specifically, repetitions 1, 3, 4, and 6 were assigned to the training set, while repetitions 2 and 5 were assigned to the test set. For DB10, additional repetitions (7 and 8) were included in the training set.

2.4.2. Model comparison

To provide a fair benchmark, four representative deep learning models were selected for comparison: Shallow CNN [13], ResNet-1D [35], Multi-Scale Convolutional Neural Network (MKCNN) [7], and Narrow Kernel Dual-view Feature Fusion Convolutional Neural Network (NKDFF) [19]. These models were selected because they achieve competitive performance on sEMG-based hand gesture recognition while representing different architectural paradigms, ranging from lightweight convolutional networks to residual and multi-scale architectures. Moreover, their implementation details are sufficiently documented to enable faithful reproduction. NKDFF also provides a multimodal variant that incorporates accelerometer data. Although additional recent models were considered, they were not included because insufficient implementation details, the absence of open-source code, or substantial differences in the original experimental protocols prevented a reliable reproduction and fair comparison.

The open-source codebase was designed to be extensible, enabling the straightforward integration of additional models and future benchmark extensions. This design choice is intended to encourage community contributions to the source code, support the expansion of the benchmark analysis, and promote fair and reproducible evaluation of novel approaches.

Traditional machine learning models were also included in the evaluation, as they remain competitive for sEMG-based hand gesture recognition [13], as discussed in Section 1 and further confirmed in Section 3. Specifically, a RF classifier and an SVM were evaluated. These models were trained using the same set of handcrafted features listed in [12]. Additional details on the features and the hyperparameter search grids are provided in the Supplementary Materials.

2.4.3. Data Augmentation

Data augmentation was incorporated during training to mitigate overfitting. Specifically, a signal warping strategy was implemented to randomly stretch or compress portions of the input signal along the temporal axis, similarly to [26, 36]. The procedure is defined as follows:

1. The input window is divided into multiple segments.
2. Up to half of these segments are randomly stretched, while the remaining segments are squeezed.
3. A non-uniform temporal grid is constructed according to the selected stretch and compression operations. The grid values are determined by the stretch and compression strength hyperparameters.
4. The input window is interpolated onto the non-uniform grid using the Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) [37].
5. The resulting signal is treated as uniformly sampled and resampled to the original temporal grid using PCHIP.

Mini-batches were augmented with a probability of 70% during training. When signal warping was applied, the number of segments (4, 6, or 8), compression strength (1.25, 1.5, or 2.0), and stretch strength (1.0, 1.5, or 2.0) were randomly sampled from a predefined hyperparameter grid. To improve computational efficiency, the same augmentation parameters were applied to all samples within a mini-batch through broadcasting along the batch dimension as well as to all input signals to preserve temporal consistency across modalities.

2.4.4. Training loss

EMG-CrossFormer was trained using a composite loss function designed to provide deep supervision across the encoding, fusion, and decoding stages. The loss jointly optimizes classification performance and embedding-space structure. More formally, let M denote the number of input modalities, and let:

- y be the ground-truth gesture label;
- $\mathbf{s}$ be the final FFN output logits;
- $\mathbf{s}_{\text{enc}_m}$ and $\mathbf{s}_{\text{fuse}}$ be auxiliary logits obtained from a linear projection of the global average pooling output of $\mathbf{z}_{\text{enc}_m}$ and $\mathbf{z}_{\text{fuse}}$, with $m \in 1, \dots, M$;
- $\mathbf{z}_{\text{dec}}$ be the query-based decoder output;
- $\mathbf{h}$ be the ground-truth handcrafted sEMG features used by machine learning models (see Supplementary Materials);
- $\hat{\mathbf{h}}$ be the handcrafted sEMG feature estimates, obtained by applying a linear predictor of the global average pooling output of $\mathbf{z}_{\text{dec}}$.

The applied training loss is defined as:

$$
\begin{aligned}
\mathcal{L}_{total} = {} & \lambda_{ce}\mathcal{L}_{CE}(\mathbf{s}, y) \\
& + \lambda_{supcon}\left[\mathcal{L}_{SupCon}(\mathbf{z}_{\text{dec}}, y) + \sum_{i=1}^{M}\mathcal{L}_{SupCon}(\mathbf{z}_{\text{enc}_i}, y)\right] \\
& + \lambda_{aux}\left[\mathcal{L}_{CE}(\mathbf{s}_{\text{fuse}}, y) + \sum_{i=1}^{M}\mathcal{L}_{CE}(\mathbf{s}_{\text{enc}_i}, y)\right] \\
& + \lambda_{hand}\mathcal{L}_{L1}(\hat{\mathbf{h}}, \mathbf{h})
\end{aligned}
\tag{4}
$$

Here, $\mathcal{L}_{CE}$ denotes the Cross-Entropy with label smoothing ($\alpha_{\text{smooth}} = 0.1$), $\mathcal{L}_{L1}$ denotes the Mean Absolute Error (L1) loss, and $\mathcal{L}_{SupCon}$ denotes the supervised contrastive loss formalized in [38]. Based on empirical tuning, the weighting hyperparameters were set to $\lambda_{ce} = 3.0$, $\lambda_{supcon} = 1.5$, $\lambda_{aux} = 1.0$, and $\lambda_{hand} = 2.0$.

2.4.5. Training Hyperparameters

EMG-CrossFormer was trained using the LAMB optimizer with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay) [39]. LAMB was preferred over the standard ADAM optimizer [40], as it provided greater training stability across all investigated models. Gradient clipping with maximum norm 0.1 was applied to further stabilize training. A standard maximum norm value of 1.0 was also tested but yielded worse results. The batch size was set to 128. The initial learning rate was set to $5.0 \cdot 10^{-4}$ for the unimodal backbones and $5.0 \cdot 10^{-5}$ for the fusion and decoder modules. An exponential scheduler with $\gamma = 0.99$ was used to decrease the learning rate after each epoch. All models were trained using mixed precision to accelerate training, reduce memory usage, and better reflect potential real-world deployment on embedded devices.

The number of epochs was set to 200 for sEMG-only training and to 100 for multimodal approaches, as multimodal training showed faster convergence. Early stopping was not adopted, since creating a separate validation set from the training gestures excessively reduced the number of available training samples. The custom training loss previously described was used to provide deep supervision during training.

All other selected deep learning models (Shallow CNN, ResNet-1D, MKCNN, and NKDFF) were trained using the same set of training hyperparameters. However, unlike EMG-CrossFormer, the learning rate was set to $5.0 \cdot 10^{-4}$ for the entire network, and categorical cross-entropy was used as the training loss. For the machine learning models, hyperparameter tuning was performed using 4-fold cross-validation on the training repetitions, followed by refitting on the entire training set using the optimal hyperparameters.

Handcrafted features were standardized using a StandardScaler fitted on the training data and incorporated into the hyperparameter search pipeline.

2.4.6. Performance Evaluation and Statistical Analysis

Model performance was evaluated using balanced accuracy, to account for class imbalance. Additional evaluation metrics, including the F1-score and Cohen's kappa, are reported in the Supplementary Materials. Pairwise model comparisons were performed using the Wilcoxon signed-rank test [41] applied to subject-level performance estimates. This non-parametric paired test matches the intra-subject design: model comparisons are computed within subjects, while the resulting subject-level differences are treated as independent observations across subjects. p-values were

corrected for multiple comparisons using the Benjamini-Hochberg method [42]. In addition to statistical significance, the mean paired improvement and its uncertainty, estimated by non-parametric bootstrap resampling, are reported to quantify the magnitude and practical relevance of performance differences.

3. Results

This section summarizes the results of nearly 3,000 training runs across different models, datasets, window lengths, and number of input modalities. In particular:

- Subsection 3.1 presents the comparison of models in the sEMG-only setting, showing that EMG-CF_{2D} performs better than the other considered models across DB2, DB3 and DB7 for all window lengths.
- Subsection 3.2 presents the performance gain obtained by adding accelerometer signals as a second input modality.
- Subsection 3.3 evaluates EMG-CrossFormer performance on DB10, where gaze data are incorporated as a third input modality.
- Subsection 3.4 compares the computational cost, memory footprint, and inference latency of different EMG-CrossFormer variants.

For each configuration (dataset, window length, and number of input modalities), the description of the results focuses on the mean paired balanced accuracy difference between the best-performing EMG-CrossFormer variant and the strongest competing model listed in subsubsection 2.4.2. Complete model-level statistical analyses for unimodal and multimodal settings, together with additional performance metrics, are provided in the Supplementary Materials.

3.1. Single modality: sEMG-only input

Table 2 summarizes the results of the selected models in the unimodal-input setting (sEMG-only). EMG-CF_{2D} achieves the highest mean balanced accuracy for every dataset and window length, with consistent but modest improvements over the strongest competing model. The closest competitors are RF and MKCNN, the latter providing the 2D multi-scale backbone used in EMG-CF_{2D}. By contrast, EMG-CF_{1D} performs below the 2D variant, suggesting that explicit spatio-temporal feature extraction is beneficial when only sEMG signals are available.

On DB2, EMG-CF_{2D} surpasses RF by 0.79 percentage points (pp) at 200 ms (CI = $[0.23, 1.31]$, $p_{FDR} = 0.003$), and MKCNN by 1.44 pp at 150 ms (CI = $[0.97, 1.85]$, $p_{FDR} < 0.001$) and 1.50 pp at 100 ms (CI = $[-0.16, 2.42]$, n.s.). On DB3, EMG-CF_{2D} surpasses RF by 1.46 pp at 200 ms (CI = $[-0.20, 3.11]$, n.s.) and 1.44 pp at 150 ms (CI = $[-0.67, 3.35]$, n.s.), and MKCNN by 1.28 pp at 100 ms (CI = $[-0.16, 2.42]$, n.s.). On DB7, EMG-CF_{2D} surpasses RF by 1.04 pp at 200 ms (CI = $[0.36, 1.69]$, $p_{FDR} = 0.006$), and MKCNN by 1.81 pp at 150 ms (CI = $[1.42, 2.20]$, $p_{FDR} < 0.001$) and by 2.09 pp at 100 ms (CI = $[1.75, 2.45]$, $p_{FDR} < 0.001$).

3.2. Two modalities: sEMG and Accelerometer inputs

Table 3 summarizes the results obtained when sEMG and accelerometer signals are jointly used as input. In this multimodal setting, the two EMG-CrossFormer variants achieve the highest mean balanced accuracy across all datasets and window lengths. EMG-CF_{2D} ranks first on DB2 and DB3, while EMG-CF_{1D} achieves the best DB7 performance at 150 ms and 200 ms. In addition, EMG-CrossFormer exhibits one of the largest improvements from the unimodal to the multimodal setting, surpassed only by NKDFF; however, NKDFF always achieves a lower absolute balanced accuracy.

On DB2, EMG-CF_{2D} surpasses NKDFF by 1.77 pp at 200 ms (CI = $[1.31, 2.29]$, $p_{FDR} < 0.001$), 2.37 pp at 150 ms (CI = $[1.94, 2.79]$, $p_{FDR} < 0.001$), and 2.88 pp at 100 ms (CI = $[2.40, 3.36]$, $p_{FDR} < 0.001$). On DB3, EMG-CF_{2D} outperforms NKDFF by 3.02 pp at 200 ms (CI = $[0.56, 5.64]$, n.s.), 4.37 pp at 150 ms (CI = $[2.31, 6.59]$, $p_{FDR} = 0.004$), and 6.39 pp at 100 ms (CI = $[4.43, 8.30]$, $p_{FDR} = 0.002$). On DB7, EMG-CF_{1D} surpasses RF by 1.05 pp at 200 ms (CI = $[0.28, 1.88]$, n.s.) and 1.61 pp at 150 ms (CI = $[0.92, 2.38]$, $p_{FDR} < 0.001$), whereas EMG-CF_{2D} surpasses RF by 2.16 pp at 100 ms (CI = $[1.40, 2.99]$ $p_{FDR} < 0.001$).

3.3. Three modalities: sEMG, Accelerometers and Gaze inputs

Table 4 and Figure 2 summarize the performance of EMG-CrossFormer variants on DB10 as the number of input modalities increases. The transition from sEMG-only to multimodal decoding led to mean paired accuracy gains of up to +20.65 pp, with the best accuracy obtained by EMG-CF_{2D} using sEMG plus accelerometers and a 100 ms window ($(92.06 \pm 3.91)\%$). However, adding gaze does not always further improve decoding accuracy over sEMG plus accelerometers, which supports prior evidence that gaze is more informative for predicting imminent movements than for decoding executed ones. This limited gain may also be related to the specific experimental setting adopted in the original DB10 study.

With a window length of 200 ms, both EMG-CF_{1D} and EMG-CF_{2D} achieved their highest mean paired accuracy in the sEMG plus accelerometers setting, with $(85.00 \pm 7.07)\%$ (+17.57 pp over sEMG-only) and $(91.69 \pm 4.38)\%$ (+18.20 pp over sEMG-only), respectively. With a window length of 150 ms, EMG-CF_{1D} achieved its highest mean paired accuracy in the three-modality setting, with $(86.40 \pm 6.38)\%$ (+18.54 pp over sEMG-only), whereas EMG-CF_{2D} achieved its top performance in the sEMG plus accelerometers setting, with $(91.80 \pm 4.30)\%$ (+18.85 pp over sEMG-only). With a window length of 100 ms, EMG-CF_{1D} achieved its highest mean paired accuracy in the three-modality setting, with $(86.19 \pm 5.19)\%$ (+20.43 pp over sEMG-only), whereas EMG-CF_{2D} achieved its top performance in the sEMG plus accelerometers setting, with $(92.06 \pm 3.91)\%$ (+20.65 pp over sEMG-only).

Table 2
Balanced Accuracy (%) across models, datasets, and window lengths using only sEMG signals

Model	DB2			DB3			DB7		
	100 ms	150 ms	200 ms	100 ms	150 ms	200 ms	100 ms	150 ms	200 ms
SVM	58.29±6.87	60.34±7.11	62.05±7.37	39.98±7.01	42.22±7.10	43.66±7.24	64.91±6.38	67.46±6.51	69.49±6.65
Random Forest	67.26±7.06	69.86±6.86	71.55±6.74	46.51±8.13	49.17±7.75	51.02±7.76	73.70±5.88	76.48±5.64	78.11±5.51
ShallowCNN	63.49±5.93	62.38±5.82	59.38±5.52	43.21±7.28	42.44±7.04	40.89±6.76	69.16±6.51	67.75±6.63	65.98±6.67
ResNet-1D	62.68±6.82	63.13±6.26	63.99±6.82	41.63±9.72	43.31±8.18	44.03±9.89	70.02±7.10	70.64±6.42	71.92±6.63
MKCNN	68.62±6.49	70.07±6.17	70.87±5.95	47.53±8.87	49.13±8.55	50.85±8.31	75.11±6.28	76.55±6.34	77.44±6.45
NKDFF	61.86±10.58	66.15±9.61	68.39±9.77	43.77±8.60	44.37±14.61	47.04±12.83	65.64±10.16	71.08±9.69	66.93±14.94
EMG-CF_{1D}	66.91±7.75	68.62±7.98	70.24±7.46	46.64±9.74	48.14±11.48	51.32±9.73	74.92±5.94	76.98±6.46	77.38±6.66
EMG-CF_{2D}	**70.12±6.69**	**71.51±6.28**	**72.33±6.13**	**48.80±9.65**	**50.61±9.80**	**52.48±8.16**	**77.19±6.32**	**78.36±6.33**	**79.16±6.47**

Table 3
Balance Accuracy (%) across models, datasets, and window lengths, using sEMG and accelerometer signals

Model	DB2			DB3			DB7		
	100 ms	150 ms	200 ms	100 ms	150 ms	200 ms	100 ms	150 ms	200 ms
SVM	78.59±5.49 (+20.31)	78.50±5.54 (+18.16)	78.71±5.67 (+16.66)	64.35±10.22 (+24.37)	64.21±9.80 (+21.99)	64.30±9.87 (+20.64)	82.63±5.73 (+17.73)	83.15±5.26 (+15.68)	83.73±5.43 (+14.24)
Random Forest	87.19±4.35 (+19.93)	87.58±4.25 (+17.72)	87.98±4.26 (+16.43)	74.39±8.98 (+27.88)	75.11±8.86 (+25.94)	75.56±8.70 (+24.55)	90.34±5.51 (+16.64)	90.75±5.20 (+14.27)	91.07±5.16 (+12.95)
NKDFF	87.78±4.03 (+25.91)	88.24±3.71 (+22.09)	88.61±3.79 (+20.22)	73.84±8.36 (+30.07)	76.03±8.62 (+31.66)	76.58±8.71 (+29.54)	89.65±5.18 (+24.01)	90.35±5.71 (+19.27)	90.61±5.36 (+23.68)
EMG-CF_{1D}	90.16±3.55 (+23.24)	90.43±3.25 (+21.81)	90.34±3.52 (+20.10)	79.11±7.88 (+32.47)	78.75±7.74 (+30.61)	78.65±9.35 (+27.33)	92.39±4.10 (+17.47)	**92.79±3.92** (+15.81)	**92.75±4.19** (+15.37)
EMG-CF_{2D}	**90.66±3.38** (+20.53)	**90.61±3.05** (+19.10)	**90.38±3.22** (+18.05)	**80.23±6.42** (+31.43)	**80.40±5.95** (+29.78)	**79.59±5.45** (+27.12)	**92.50±4.18** (+15.31)	92.36±4.12 (+14.00)	92.11±4.29 (+12.96)

Parenthesized values indicate the mean balanced accuracy paired difference (pp) relative to the corresponding sEMG-only setting.

Table 4 further stratifies EMG-CF accuracies by healthy and amputee subject groups. Multimodal integration narrowed the performance gap between the two groups, most clearly for EMG-CF_{2D}: the healthy–amputee difference decreased from approximately 15 pp with sEMG alone to 3.8–4.8 pp with sEMG plus accelerometers across window lengths, although mean accuracy remained lower in amputees.

3.4. Computational analysis

Table 5 summarizes the computational cost, memory footprint, and latency of both EMG-CF_{1D} and EMG-CF_{2D} under unimodal and multimodal configurations. Performance metrics were benchmarked on a single NVIDIA A30 GPU using CUDA 12.2 and PyTorch 2.12.0+cu126. Reported values correspond to the average profiling statistics collected over 1000 independent inference runs with a batch size of one. Results are intended as reference benchmarks, as embedded processors used in prosthetic systems generally provide significantly lower computational capabilities than the evaluation hardware.

The results demonstrate that EMG-CF_{1D} achieves the best trade-off between predictive performance and inference latency, particularly in the multimodal setting. First, the multimodal EMG-CF_{1D} model requires 5.8× fewer FLOPs than its 2D counterpart. This reduction is associated to the lightweight and efficient design of the 1D depthwise convolutional backbone, which scales more effectively with the number of input channels. Second, the model's compiled version (through TorchInductor) yields an 8.98× latency reduction for the 1D variant compared to the compiled 2D-backbone configuration under identical multimodal input. EMG-CF_{1D} is the only model achieving sub-millisecond inference latency (0.77 ± 0.10 ms) on the tested hardware.

Although EMG-CF_{2D} achieves higher classification accuracy across different datasets and window lengths, this performance gap becomes smaller in the multimodal setting. This similarity in multimodal decoding accuracy, coupled with the lower computational overhead, supports the selection of the EMG-CF_{1D} variant for real-time, resource-constrained edge applications such as myoelectric prosthetic control systems.

4. Discussion

Decoding complex hand movements from sEMG signals remains a challenging task. To achieve clinical and practical viability, deep learning architectures must be designed to handle the unique characteristics of this modality. In particular, models must extract informative representations from short temporal windows to maintain low latency. These features must effectively capture spatio-temporal relationships that characterize distinct motor patterns, overcoming the low signal-to-noise ratio intrinsic to sEMG data.

EMG-CrossFormer was developed to address several limitations of conventional unimodal convolutional architectures and to facilitate multimodal data integration in sEMG architectures. By leveraging a 2D multi-scale backbone capable of modeling localized spatio-temporal dynamics during feature extraction, the proposed model achieved superior performance across multiple databases and temporal window configurations within an sEMG-only baseline

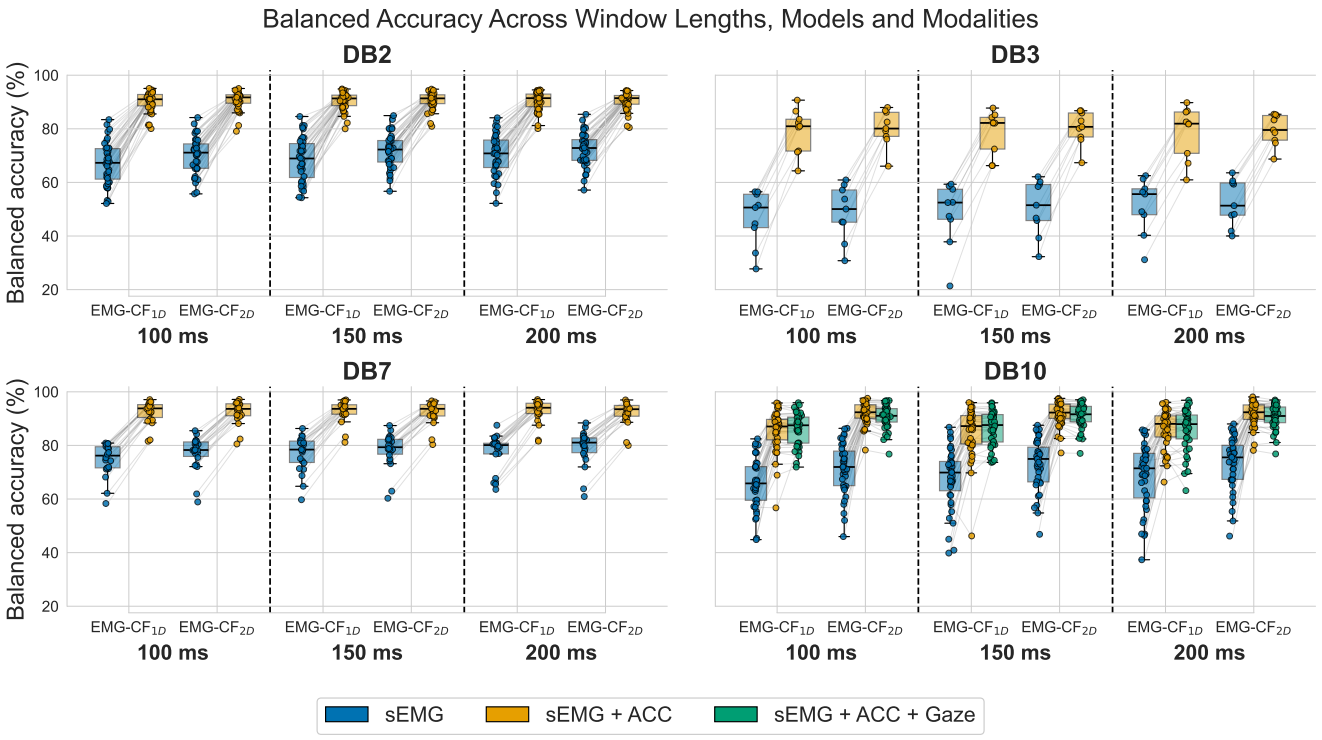

Figure 2: Hand-movement decoding performance of EMG-CrossFormer across datasets, window lengths, model variants, and input modalities. Balanced accuracy is shown for EMG-CF$_{1D}$, based on a 1D depthwise convolutional backbone, and EMG-CF$_{2D}$, based on a 2D multi-scale convolutional backbone. Results are stratified by window length (100 ms, 150 ms, and 200 ms) and input configuration: sEMG-only decoding is shown in blue, sEMG + ACC decoding is shown in orange, sEMG + ACC + Gaze in green. Each point represents one subject, and gray lines connect paired unimodal and multimodal results from the same subject. Subplots correspond to NinaPro DB2, DB3, DB7, and DB10.

Table 4
Balance Accuracy (%) of EMG-CrossFormer variants on DB10, across window lengths and input modalities.

Model	Modality*	DB10 100 ms Global	100 ms Healthy	100 ms Amputees	DB10 150 ms Global	150 ms Healthy	150 ms Amputees	DB10 200 ms Global	200 ms Healthy	200 ms Amputees
EMG-CF$_{1D}$	S	65.76±9.23	69.16±7.60	57.13±7.14	67.86±11.14	71.49±9.74	58.62±8.93	68.41±11.94	72.81±9.48	57.22±10.12
	S + A	85.54±7.64	88.72±4.62	77.44±7.81	85.11±9.09	88.62±5.27	76.18±10.55	**85.98±7.07**	**88.77±5.29**	**78.89±6.00**
	S + A + G	**86.19±5.99**	**88.84±4.46**	**79.44±3.65**	**86.40±6.34**	**89.05±4.98**	**79.66±4.00**	85.76±7.76	88.51±6.33	78.77±6.61
EMG-CF$_{2D}$	S	71.41±9.44	75.58±6.54	60.78±7.08	72.95±9.37	77.08±6.49	62.43±7.06	73.49±9.71	77.79±6.31	62.54±8.11
	S + A	**92.06±3.91**	**93.12±3.70**	**89.34±3.02**	**91.80±4.30**	**93.08±3.89**	**88.54±3.47**	**91.70±4.38**	**93.05±3.88**	**88.25±3.65**
	S + A + G	90.65±4.40	91.92±3.91	87.40±3.89	90.86±4.52	92.21±4.05	87.43±3.78	90.86±4.43	92.22±3.93	87.38±3.64

*S stands for sEMG, A stands for Accelerometer, G stands for Gaze

setup. Although performance gains are statistically significant according to FDR-corrected signed-rank tests, the mean paired accuracy gap between EMG-CrossFormer and other competitive approaches, such as RFs or MKCNN, remains modest when considering the observed variability of results. Furthermore, sEMG-only decoding accuracy for trans-radial amputees (DB3) degrades by 20 pp or more compared to intact individuals (DB2 and DB7). These findings suggest that sEMG signals alone may be insufficient to reliably discriminate among a large number of partially correlated hand movements. Consequently, integrating additional input modalities may help distinguish between similar movements by combining complementary information related to both motor intent and the physical execution of movement. However, the effectiveness of multimodal systems strongly depends on the design of the fusion strategy, which must enable the model to fully exploit the information contained in each modality.

Most existing multimodal deep learning approaches, such as NKDFF, combine modality-specific representations through feature concatenation before the final classification stage. While straightforward, this strategy limits the model's ability to learn complex interactions between modalities. In practice, concatenation may encourage the network to rely predominantly on the most informative modality while underutilizing the complementary information provided by the others. EMG-CrossFormer addresses this limitation through a dedicated fusion module based on cross-attention mechanisms. By integrating multimodal interactions before

Table 5
Computational analysis of EMG-CrossFormer Variants

Model	Input	Params (M)	FLOPs (M)	Configuration	Latency (ms)	Memory (Mb)
EMG-CF_{1D}	sEMG [8×200] [4×200]	1.03	54.40	FP32 CPU	5.52±0.26	-
				FP32 GPU	3.27±0.09	16.42
				FP16 GPU	3.62±0.44	12.30
				Compiled	0.74±0.06	8.06
	sEMG [12×200] ACC [36×200]	1.07	59.04	FP32 CPU	6.15±0.38	-
				FP32 GPU	3.53±0.23	16.73
				FP16 GPU	3.93±0.44	12.46
				Compiled	0.77±0.10	8.37
EMG-CF_{2D}	sEMG [8×200] [4×200]	1.20	79.18	FP32 CPU	12.26±0.22	-
				FP32 GPU	6.59±0.59	17.97
				FP16 GPU	6.91±0.72	13.11
				Compiled	5.84±0.55	17.98
	sEMG [12×200] ACC [36×200]	1.22	342.44	FP32 CPU	15.18±1.68	-
				FP32 GPU	7.55±0.64	19.55
				FP16 GPU	7.89±0.44	13.89
				Compiled	6.92±0.47	19.57

the transformer decoder stage, the model learns representations in which each modality can condition its features on information extracted from the others. This design promotes the learning of richer cross-modal relationships that cannot be captured through simple feature aggregation. Furthermore, the supervised contrastive component of the loss function described in section 2 encourages modality-specific representations to be projected into a shared embedding space while preserving class separability, facilitating more effective multimodal integration.

Thanks to this design, EMG-CrossFormer achieves one of the highest performance improvements when switching from unimodal to a multimodal setup, even if its unimodal performance in subsection 3.1 is already superior to other models. In particular, mean paired accuracy on amputees (DB3) improves by 31.43 pp when using the shortest window length (100 ms). This improvements reduce the observed differences in decoding accuracy between intact subjects and trans-radial amputees. Furthermore, the accuracy gap between EMG-CrossFormer and the closest-performing model on the same dataset and window length increases from +1.27 pp in the unimodal setting (against MKCNN) to +6.37 pp (against NKDFF). Such an increase confirms the improved ability of the model to combine information from different signals, even for amputees, who are the targets for real-world prosthetic applications.

Beyond the overall benefit of multimodal fusion, the choice of the additional modality also represents an important practical factor. Among the investigated configurations, the inclusion of eye-tracking information leads to improvements in decoding performance for EMG-CF_{1D}, suggesting that gaze can provide discriminative cues related to motor intention. However, this improvement comes at the cost of increased prosthetic-system complexity, since eye tracking requires dedicated hardware, calibration procedures, and integration within a wearable or clinically usable setup. A similar trade-off has been reported for video-based modalities, whose integration can improve decoding accuracy but generally increases computational cost due to image acquisition and processing requirements [43]. Conversely, accelerometer signals provide complementary information related to the physical execution of movement with a comparatively simpler sensing configuration. Therefore, the selection of auxiliary modalities should consider not only decoding accuracy, but also hardware complexity, computational cost, usability, and translational feasibility for real-world prosthetic applications.

Despite these promising results, several limitations should be acknowledged. Although EMG-CrossFormer achieves superior performance while maintaining reduced FLOPs, memory footprint, and inference latency (see Table 5), it remains a black box model. In rehabilitation and other biomedical applications, interpretability is essential for improving model reliability, understanding failure scenarios, and increasing robustness to out-of-distribution samples. The 1D depthwise convolutional encoder adopted in EMG-CF_{1D} extracts channel-specific features characterizing local portions of the input signal. Still, the physiological meaning of these learned representations, as well as the temporal patterns that activate them, remains unclear. Future work should investigate architectures and analysis methods that improve the interpretability of unimodal representations. Interpretable modality-specific features would also facilitate the analysis of multimodal interactions by enabling inspection of cross-attention matrices and quantification of attention flow through transformer layers using explainable artificial intelligence (XAI) techniques such as attention rollout [44].

A second limitation concerns benchmarking and reproducibility. Although the results presented in section 3 demonstrate that EMG-CrossFormer consistently ranks first among the top-performing models across all investigated datasets and configurations, comparisons between studies remain challenging. As anticipated in section 1, differences in preprocessing pipelines, train-test splitting strategies, hyperparameter optimization procedures, and evaluation protocols can substantially influence reported performance. Furthermore, the lack of implementation details and open repositories identified in many studies prevents the inclusion of other models in the presented benchmarking. As the number of deep learning studies for hand gesture decoding continues to grow, there is an increasing need for standardized benchmarking frameworks that facilitate fair comparisons and fully reproducible evaluations. The source code made openly available in this study enables an easy integration of different models in the same experimental setting described in section 2. The community is therefore encouraged to add more models and support the design of an open benchmarking library.

Finally, the experiments were conducted primarily on the NinaPro database, which represents one of the largest and most widely adopted resource for hand movement decoding. Nevertheless, both the number of subjects and the diversity of modalities remain limited for large multimodal deep learning applications. The acquisition and public release of larger multimodal datasets could accelerate progress in

this field by enabling more comprehensive evaluations of emerging architectures. Such datasets may also support the development of foundation models for sEMG analysis that can be efficiently adapted to potential end-users through zero-shot, few-shot, or transfer-learning strategies.

5. Conclusion

This work confirms that multimodal data fusion is a powerful approach for improving unimodal sEMG-based hand gesture recognition while requiring only a limited increase in computational workload. To this end, EMG-CrossFormer is proposed as an end-to-end hybrid convolutional-transformer model for the seamless integration of multiple signals. EMG-CrossFormer combines representations from an arbitrary number of unimodal encoders through cascaded cross-attention layers, and decodes the fused representations using learnable gesture queries. The architecture was designed to allow researchers to easily customize its modules while maintaining a computational footprint suitable for embedded deployment. Unimodal backbones may differ across modalities, and the number of modalities can vary depending on the target application. EMG-CrossFormer was evaluated on four NinaPro databases (DB2, DB3, DB7, and DB10) and benchmarked against six state-of-the-art models. When trained using only sEMG signals, EMG-CrossFormer with a 2D multi-scale backbone surpassed other competing models in decoding accuracy, although the improvements were modest. In the multimodal setting, EMG-CrossFormer's decoding accuracy increased significantly for both intact and amputee subjects, maintaining the best overall performance and further widening the accuracy gap relative to the other models. These findings demonstrate that modern deep learning architectures designed for multimodal data integration can effectively leverage complementary physiological information to improve hand gesture decoding performance. However, the development of solutions that can be safely and reliably deployed in real-world assistive devices remains an open challenge for the scientific community.

CRediT authorship contribution statement

Federico Del Pup: Conceptualization, Methodology, Software, Formal Analysis, Writing - Original Draft . **Elisa Tentori:** Formal Analysis, Software, Visualization, Investigation, Writing - review and editing . **Manfredo Atzori:** Supervision, Funding acquisition, Project administration, Writing - review and editing .

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and data availability

The code used to produce both results and figures is openly available at https://github.com/DeepPNCLab/emg-crossformer. All data that support the findings of this study are openly available at https://ninapro.hevs.ch.

Acknowledgment

This document is the result of the research project funded by the European Unions Horizon Europe research and innovation programme under Grant agreement no 101137074 - HEREDITARY.

References

[1] N. Jiang, C. Chen, J. He, J. Meng, L. Pan, S. Su, X. Zhu, Bio-robotics research for non-invasive myoelectric neural interfaces for upper-limb prosthetic control: a 10-year perspective review, Natl. Sci. Rev. 10 (5) (2023) nwad048. doi:10.1093/nsr/nwad048.

[2] X. Lv, C. Dai, H. Liu, Y. Tian, L. Chen, Y. Lang, R. Tang, J. He, Gesture recognition based on sEMG using multi-attention mechanism for remote control, Neural Comput. Appl. 35 (19) (2023) 13839–13849. doi:10.1007/s00521-021-06729-6.

[3] A. Marinelli, N. Boccardo, F. Tessari, D. Di Domenico, G. Caserta, M. Canepa, G. Gini, G. Barresi, M. Laffranchi, L. De Michieli, M. Semprini, Active upper limb prostheses: a review on current state and upcoming breakthroughs, Prog. Biomed. Eng. 5 (1) (2023) 012001. doi:10.1088/2516-1091/acac57.

[4] N. Jiang, S. Dosen, K.-R. Muller, D. Farina, Myoelectric control of artificial limbs—is there a need to change focus? [in the spotlight], IEEE Signal Process. Mag. 29 (5) (2012) 152–150. doi:10.1109/MSP.2012.2203480.

[5] B. Abdikenov, D. Zholtayev, K. Suleimenov, N. Assan, K. Ozhikenov, A. Ozhikenova, N. Nadirov, A. Kapsalyamov, Emerging frontiers in robotic upper-limb prostheses: Mechanisms, materials, tactile sensors and machine learning-based EMG control: A comprehensive review, Sensors 25 (13) (2025). doi:10.3390/s25133892.

[6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[7] R. Fratti, N. Marini, M. Atzori, H. Müller, C. Tiengo, F. Bassetto, A multi-scale CNN for transfer learning in sEMG-based hand gesture recognition for prosthetic devices, Sensors 24 (22) (2024). doi:10.3390/s24227147.

[8] N. K. Karnam, S. R. Dubey, A. C. Turlapaty, B. Gokaraju, EMGHandNet: A hybrid CNN and Bi-LSTM architecture for hand activity classification using surface EMG signals, Biocybern. Biomed. Eng. 42 (1) (2022) 325–340. doi:10.1016/j.bbe.2022.02.005.

[9] M. Jabbari, R. N. Khushaba, K. Nazarpour, EMG-based hand gesture classification with long short-term memory deep recurrent neural networks, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 3302–3305. doi:10.1109/EMBC44109.2020.9175279.

[10] S. Zabihi, E. Rahimian, A. Asif, A. Mohammadi, TraHGR: Transformer for hand gesture recognition via electromyography, IEEE Trans. Neural Syst. Rehabil. Eng. 31 (2023) 4211–4224. doi:10.1109/TNSRE.2023.3324252.

[11] F. Stival, S. Michieletto, M. Cognolato, E. Pagello, H. Müller, M. Atzori, A quantitative taxonomy of human hand grasps, J. NeuroEng. Rehabil. 16 (1) (2019) 28. doi:10.1186/s12984-019-0488-x.

[12] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, H. Müller, Electromyography data for non-invasive naturally-controlled robotic hand prostheses, Sci. Data 1 (1) (2014) 140053. doi:10.1038/sdata.2014.53.

[13] M. Atzori, M. Cognolato, H. Müller, Deep learning with convolutional neural networks applied to electromyography data: A resource

for the classification of movements for prosthetic hands, Front. Neurorobotics 10 (2016). doi:10.3389/fnbot.2016.00009.

[14] M. Atzori, A. Gijsberts, S. Heynen, A.-G. M. Hager, O. Deriaz, P. van der Smagt, C. Castellini, B. Caputo, H. Müller, Building the Ninapro database: A resource for the biorobotics community, in: 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2012, pp. 1258–1265. doi:10.1109/BioRob.2012.6290287.

[15] T. R. Farrell, R. F. Weir, The optimal controller delay for myoelectric prostheses, IEEE Trans. Neural Syst. Rehabil. Eng. 15 (1) (2007) 111–118. doi:10.1109/TNSRE.2007.891391.

[16] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, G. Rasool, Transformers in time-series analysis: A tutorial, Circuits Syst. Signal Process. 42 (12) (2023) 7433–7466. doi:10.1007/s00034-023-02454-8.

[17] A. Krasoulis, I. Kyranou, M. S. Erden, K. Nazarpour, S. Vijayakumar, Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements, J. NeuroEng. Rehabil. 14 (1) (2017) 71. doi:10.1186/s12984-017-0284-4.

[18] M. Atzori, A. Gijsberts, H. Müller, B. Caputo, Classification of hand movements in amputated subjects by sEMG and accelerometers, in: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2014, pp. 3545–3549. doi:10.1109/EMBC.2014.6944388.

[19] B. Jiang, H. Wu, Q. Xia, G. Li, H. Xiao, Y. Zhao, NKDFF-CNN: A convolutional neural network with narrow kernel and dual-view feature fusion for multitype gesture recognition based on sEMG, Digit. Signal Process. 156 (2025) 104772. doi:10.1016/j.dsp.2024.104772.

[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 213–229. doi:10.1007/978-3-030-58452-8_13.

[21] M. Cognolato, A. Gijsberts, V. Gregori, G. Saetta, K. Giacomino, A.-G. M. Hager, A. Gigli, D. Faccio, C. Tiengo, F. Bassetto, B. Caputo, P. Brugger, M. Atzori, H. Müller, Gaze, visual, myoelectric, and inertial data of grasps for intelligent prosthetics, Sci. Data 7 (1) (2020) 43. doi:10.1038/s41597-020-0380-3.

[22] C. J. De Luca, L. Donald Gilmore, M. Kuznetsov, S. H. Roy, Filtering the surface EMG signal: Movement artifact and baseline noise contamination, J. Biomech. 43 (8) (2010) 1573–1579. doi:10.1016/j.jbiomech.2010.01.027.

[23] F. R. Hampel, The influence curve and its role in robust estimation, Journal of the American Statistical Association 69 (346) (1974) 383–393. doi:10.1080/01621459.1974.10482962.

[24] J. W. Grootjen, H. Weingärtner, S. Mayer, Uncovering and addressing blink-related challenges in using eye tracking for interactive systems, in: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, CHI '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 23. doi:10.1145/3613904.3642086.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Proc. Adv. Neural Inf. Process. Syst. 32 (2019). doi:10.48550/arXiv.1912.01703.

[26] F. Del Pup, R. Brun, F. Iotti, E. Paccagnella, M. Pezzato, S. Bertozzo, A. Zanola, L. F. Tshimanga, H. Müller, M. Atzori, TransformEEG: Towards improving model generalizability in deep learning-based EEG Parkinson's disease detection, Neurocomputing 664 (2026) 132075. doi:10.1016/j.neucom.2025.132075.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, pp. 5998–6008. doi:10.48550/arXiv.1706.03762.

[28] N. Shazeer, GLU variants improve transformer, arXiv Preprint (Feb 2020). doi:10.48550/arXiv.2002.05202.

[29] K. Zhao, Z. Zhang, H. Wen, B. Liu, J. Li, A. d'Avella, A. Scano, Muscle synergies for evaluating upper limb in clinical applications: A systematic review, Heliyon 9 (5) (May 2023). doi:10.1016/j.heliyon.2023.e16202.

[30] C. Brambilla, M. Atzori, H. Müller, A. d'Avella, A. Scano, Spatial and temporal muscle synergies provide a dual characterization of low-dimensional and intermittent control of upper-limb movements, Neuroscience 514 (2023) 100–122. doi:10.1016/j.neuroscience.2023.01.017.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
URL http://jmlr.org/papers/v12/pedregosa11a.html

[32] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., SciPy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (3) (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[33] S. Seabold, J. Perktold, statsmodels: Econometric and statistical modeling with python, in: 9th Python in Science Conference, 2010, pp. 92–96. doi:10.25080/Majora-92bf1922-011.

[34] M. L. Waskom, Seaborn: statistical data visualization, J. Open Source Softw. 6 (60) (2021) 3021. doi:10.21105/joss.03021.

[35] R. Ganiga, M. S. N., W. Choi, S. Pan, ResNet1D-based personal identification with multi-session surface electromyography for electronic health record integration, Sensors 24 (10) (2024). doi:10.3390/s24103140.

[36] A. Saeed, V. Ungureanu, B. Gfeller, Sense and learn: Self-supervision for omnipresent sensors, Mach. Learn. Appl. 6 (2021) 100152. doi:10.1016/j.mlwa.2021.100152.

[37] F. N. Fritsch, J. Butland, A method for constructing local monotone piecewise cubic interpolants, SIAM J. Sci. Stat. Comput. 5 (2) (1984) 300–304. doi:10.1137/0905021.

[38] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, in: Advances in neural information processing systems, Vol. 33, Curran Associates, Inc., 2020, pp. 18661–18673. doi:10.48550/arXiv.2004.11362.

[39] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.-J. Hsieh, Large batch optimization for deep learning: Training BERT in 76 minutes, in: International Conference on Learning Representations, 2020, p. 17. doi:10.48550/arXiv.1904.00962.

[40] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proc. Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, 2015, p. 13. doi:10.48550/arXiv.1412.6980.

[41] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83. doi:10.2307/3001968.

[42] Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: A practical and powerful approach to multiple testing, J. R. Stat. Soc., B (Methodol.) 57 (1) (1995) 289–300.
URL http://www.jstor.org/stable/2346101

[43] L. T. Taverne, M. Cognolato, T. Bützer, R. Gassert, O. Hilliges, Video-based prediction of hand-grasp preshaping with application to prosthesis control, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 4975–4982. doi:10.1109/ICRA.2019.8794175.

[44] S. Abnar, W. Zuidema, Quantifying attention flow in transformers, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 4190–4197. doi:10.18653/v1/2020.acl-main.385.

A. Supplementary Materials

This document provides supplementary materials for the research work Multimodal Surface EMG Hand Gesture Recognition Using Query-Based Transformers for Prosthetic Control. Specifically, it complements the methodology described in Section II of the main text and further supports the results presented in Section III.

A.1. Further details on model architecture

Figure 1 expands on the EMG-CrossFormer architecture, reporting the structure of the network modules used to train the model, the unimodal backbones, and the initialization hyperparameters.

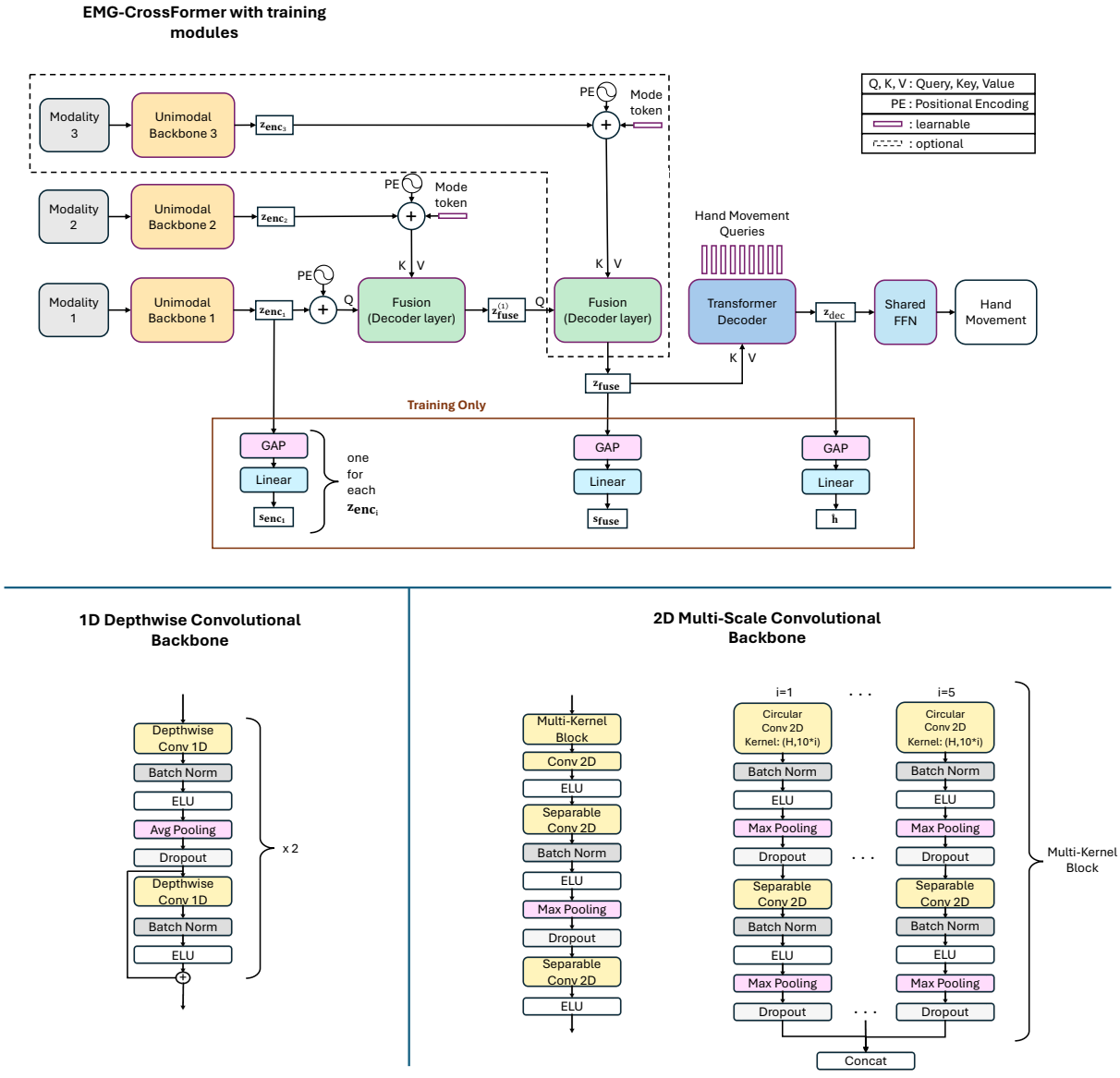

Figure 1: Detailed representation of the EMG-CrossFormer architecture. The model combines representations ($\mathbf{z}_{\text{enc}_i}$) from an arbitrary number of unimodal backbones through cascaded fusion layers (transformer decoder layers). The transformer decoder block decodes the fused representations ($\mathbf{z}_{\text{fuse}}$) using learnable hand-movement queries. A final shared FFN outputs hand-movement predictions using the decoded representations ($\mathbf{z}_{\text{dec}}$). During training, auxiliary predictions were generated from intermediate representations by applying Global Average Pooling (GAP), followed by a linear projection layer.

1D Depthwise Convolutional Backbone. This backbone consists of two stacked depthwise convolutional blocks. Each block applies a depthwise 1D convolution, batch normalization, ELU activation, average pooling, and dropout, followed by a second depthwise convolution, batch normalization, and ELU activation. A residual connection adds the feature representation immediately after dropout to the output of the second convolutional stage. The output of the second block is projected to the EMG-CrossFormer embedding dimension through a two-layer FFN with hidden size 128 and ReLU activation.

Table 1: Hyperparameters of the 1D depthwise backbone.

Parameter	Value
First depthwise multiplier	2
Second depthwise multiplier	1
Kernel length	41
Average pooling kernel size	2
Average pooling stride	2
Dropout rate	0.025
ELU α	0.1
BatchNorm momentum	0.25

2D Multi-Scale Convolutional Backbone.
This backbone combines parallel 2D convolutional branches that extract temporal features at different scales (referred to as the *multi-kernel block*), followed by a shared feature-mixing branch composed of separable 2D convolutions.
As schematized in Figure 1, the input signal is first reshaped into a single-channel pseudo-image by introducing a channel dimension. The resulting representation is processed by a multi-kernel block consisting of five parallel branches. Each branch applies a 2D convolution with circular padding to account for the spatial arrangement of forearm electrodes, followed by batch normalization, ELU activation, max pooling, dropout, a separable 2D convolution, and a second sequence of batch normalization, ELU activation, max pooling, and dropout.

The outputs of the five branches are concatenated and passed to a shared feature-mixing branch composed of a 1×1 convolution, ELU activation, a separable 2D convolution followed by batch normalization and ELU activation, adaptive max pooling, dropout, a final separable 2D convolution, and ELU activation. The final representation is obtained by flattening the last two dimensions and projecting it to the EMG-CrossFormer embedding dimension through a linear layer.

Initialization hyperparameters are provided in the openly available source code. The parameters of the multi-kernel block and shared feature-mixing branch are reported in Tables 2 and 3, respectively.

Table 2: Hyperparameters of the multi-kernel block.

Parameter	Value
Number of parallel branches	5
Output channels (first convolution)	32
Kernel size	$C \times (10i)$, $i \in \{1, 2, 3, 4, 5\}$
Channel dimension C	3 (sEMG), 9 (accelerometers), 1 (gaze)
Circular padding	1 (sEMG), 3 (accelerometers), 0 (gaze)
Circular padding application	Forearm-channel dimension only
First max pooling kernel size	$[1, 20]$
First max pooling stride	1
Separable convolution output channels	64
Separable convolution depthwise multiplier	1
Separable convolution kernel size	3×3
Dropout rate	0.2
Second max pooling kernel size	$[2, 2]$
Second max pooling stride	1

Table 3: Hyperparameters of the shared feature-mixing branch.

Parameter	Value
1×1 convolution output channels	128
Separable convolution output channels	128
Separable convolution depthwise multiplier	1
Separable convolution kernel size	3×3
Adaptive max pooling output size	$(5, 2)$
Dropout rate	0.2

Transformer Decoder.
The decoder follows the default PyTorch implementation and consists of four decoder layers. Each layer uses an embedding dimension of 128, eight attention heads, a feedforward hidden dimension of 128, no dropout, and a SwiGLU activation function.

Table 4: Transformer decoder hyperparameters.

Parameter	Value
Number of decoder layers	4
Embedding dimension	128
Number of attention heads	8
Feedforward hidden dimension	128
Dropout rate	0.0
Activation function	SwiGLU

Train-only Layers.
During training, auxiliary predictions were generated from intermediate representations by applying Global Average Pooling (GAP), followed by a linear projection layer.

A.2. Further details on the Machine Learning pipeline

A set of handcrafted features commonly used in prior work was extracted and used to train the machine learning models. Specifically, the following 29 features were extracted from each channel:

- Root Mean Square.
- Mean Absolute Value.
- Waveform Length: cumulative length of the waveform, computed as the sum of the absolute differences between consecutive samples.
- Zero Crossings: number of sign changes in the signal.
- Slope Sign Changes: number of sign changes in the first derivative of the signal.
- Histogram Features: counts of samples that fall into each bin. The signal is quantized into 20 bins defined over the range $[-3\sigma, 3\sigma]$, producing 20 features per channel.
- Marginal Discrete Wavelet Transform: computed using a Daubechies-7 (`db7`) wavelet decomposition with 3 levels, producing 4 features per channel.

The implementation of the feature extraction procedure is available in the open-source code repository.
Machine learning hyperparameter tuning was performed using 4-fold cross-validation on the training repetitions, followed by refitting on the entire training set using the identified optimal set of hyperparameters. Table 5 lists all optimized hyperparameters and the value grid. Due to the long training time required to test all hyperparameter combinations, a preliminary screening was performed to identify the most suitable sub-grid and reduce the overall training time.

Table 5: Machine Learning model hyperparameters.

Model	Parameter	Values
Random Forest	Number of estimators	100, 200, 500
	maximum depth	None, 10, 20, 30
	minimum samples to split a node	2, 5, 10
SVM	kernel	poly, rbf
	C	0.0001, 0.001, 0.01, 0.1, 1.0
	γ	0.0001, 0.001, 0.01, 0.1, 1.0

A.3. Further result visualization and statistical analysis

A.3.1. sEMG-only setting

Each EMG-CrossFormer variant was compared with each competing model using subject-level balanced accuracies paired by subject (Figure 2 and Figure 3). Comparisons were performed separately for each database and window length, yielding 108 tests: 3 databases × 3 window lengths × 2 EMG-CrossFormer variants × 6 competing models. For each comparison, we computed the mean paired difference in balanced accuracy (EMG-CF minus competitor, in percentage points) and its 95% bootstrap confidence interval by resampling the subject-level differences 10,000 times. Statistical significance was assessed using a one-sided paired Wilcoxon signed-rank test, with the alternative hypothesis that EMG-CrossFormer achieved higher balanced accuracy. The resulting 108 p-values were jointly adjusted using the Benjamini–Hochberg FDR procedure, with adjusted $p < 0.01$ considered significant.

EMG-CF_{2D} showed the most consistent advantage (Figure 2). In DB2 and DB7, it significantly outperformed all competing models at all window lengths. In DB3, it significantly outperformed SVM, ShallowCNN, and ResNet-1D at all window lengths. Comparisons with NKDFF and MKCNN reached significance at only one window length each, while no comparison with Random Forest was significant.

EMG-CF_{1D} showed a weaker and less consistent pattern (Figure 3). Significant improvements over SVM, ShallowCNN, and ResNet-1D were frequent in DB2 and DB7, but less consistent in DB3 and against NKDFF. EMG-CF_{1D} did not significantly outperform MKCNN in any configuration and significantly outperformed Random Forest only in DB7 at 100 ms.

A.3.2. sEMG + ACC setting

We next tested whether EMG-CrossFormer retained its advantage when inertial information was added to the sEMG input. In this multimodal setting, each EMG-CrossFormer variant was compared with the competing models available for sEMG + ACC decoding: SVM, NKDFF, and Random Forest. Comparisons were performed separately for each database and window length, yielding 54 tests: 3 databases × 3 window lengths × 2 EMG-CrossFormer variants × 3 competing models. For each comparison, we computed the mean paired difference in balanced accuracy (EMG-CF minus competitor, in percentage points) and its 95% bootstrap confidence interval by resampling the subject-level differences 10,000 times. Statistical significance was assessed using a one-sided paired Wilcoxon signed-rank test. The resulting 54 p-values were jointly adjusted using the Benjamini–Hochberg FDR procedure, with adjusted $p < 0.01$ considered significant.

Both EMG-CrossFormer variants showed strong multimodal performance. EMG-CF_{1D} significantly outperformed all three competing models at all window lengths in DB2 and DB7 (Figure 5). In DB3, it significantly outperformed all competitors at 100 ms, whereas only the comparisons with SVM remained significant at 150 and 200 ms. EMG-CF_{2D} significantly outperformed all competing models at all window lengths in DB2 (Figure 4). In DB7, all comparisons were significant except that with Random Forest at 200 ms. In DB3, all comparisons were significant at 100 and 150 ms, whereas only the comparison with SVM remained significant at 200 ms.

Overall, the advantage of EMG-CrossFormer in the multimodal sEMG + ACC setting was highly consistent in DB2 and DB7 and less consistent in DB3, particularly at longer window lengths against NKDFF and Random Forest.

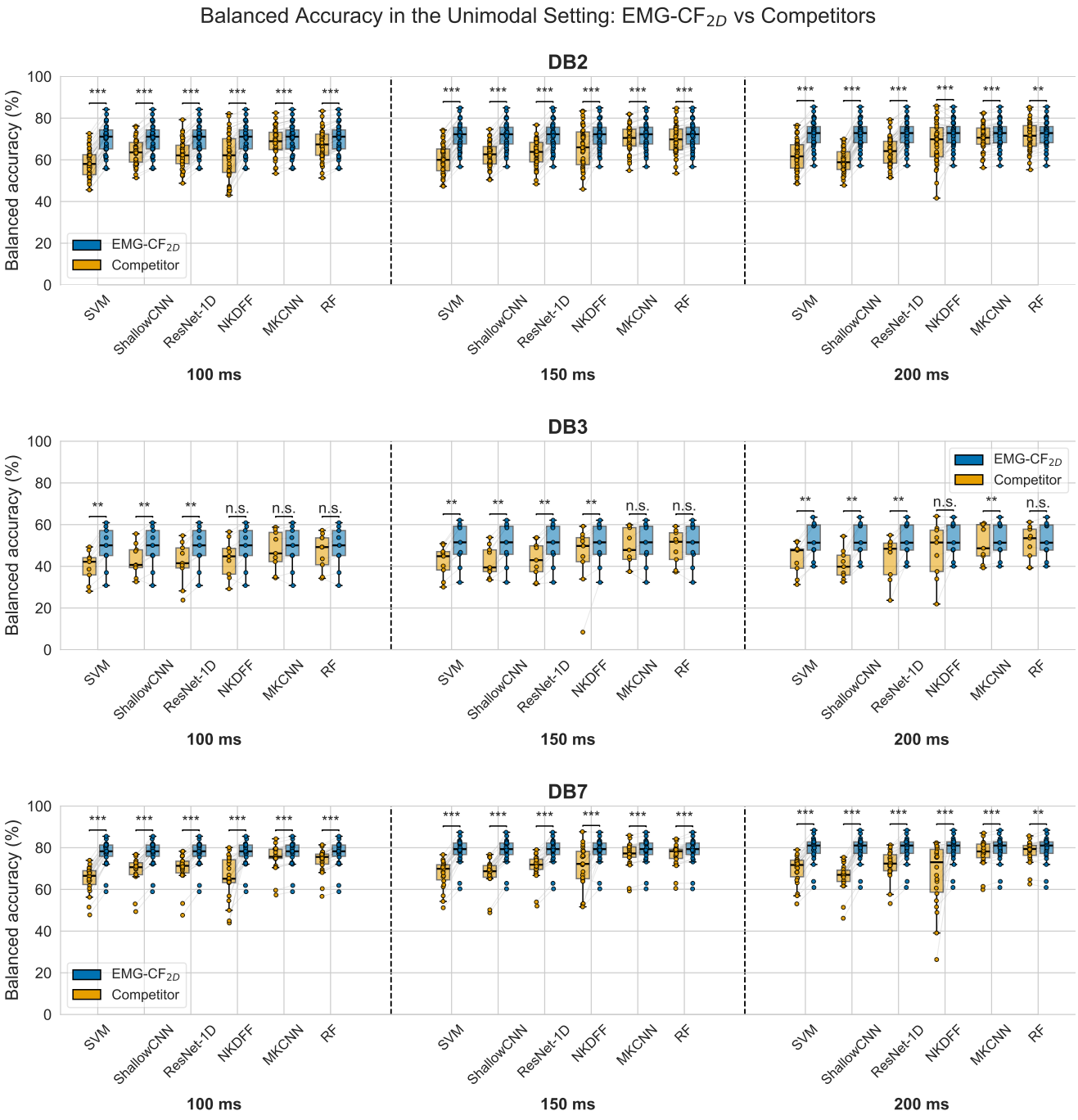

Figure 2: sEMG-only paired comparisons between EMG-CF_{2D} and the competing models for NinaPro DB2, DB3, and DB7 across 100, 150, and 200 ms windows. Boxplots summarize the subject-level balanced accuracies; points represent individual subjects, and gray lines connect paired observations. Asterisks indicate one-sided paired Wilcoxon signed-rank tests adjusted across the 108 unimodal comparisons using the Benjamini-Hochberg FDR procedure: $^{***}p_{\text{FDR}} < 0.001$ and $^{**}p_{\text{FDR}} < 0.01$. Comparisons with $p_{\text{FDR}} \geq 0.01$ are denoted n.s.

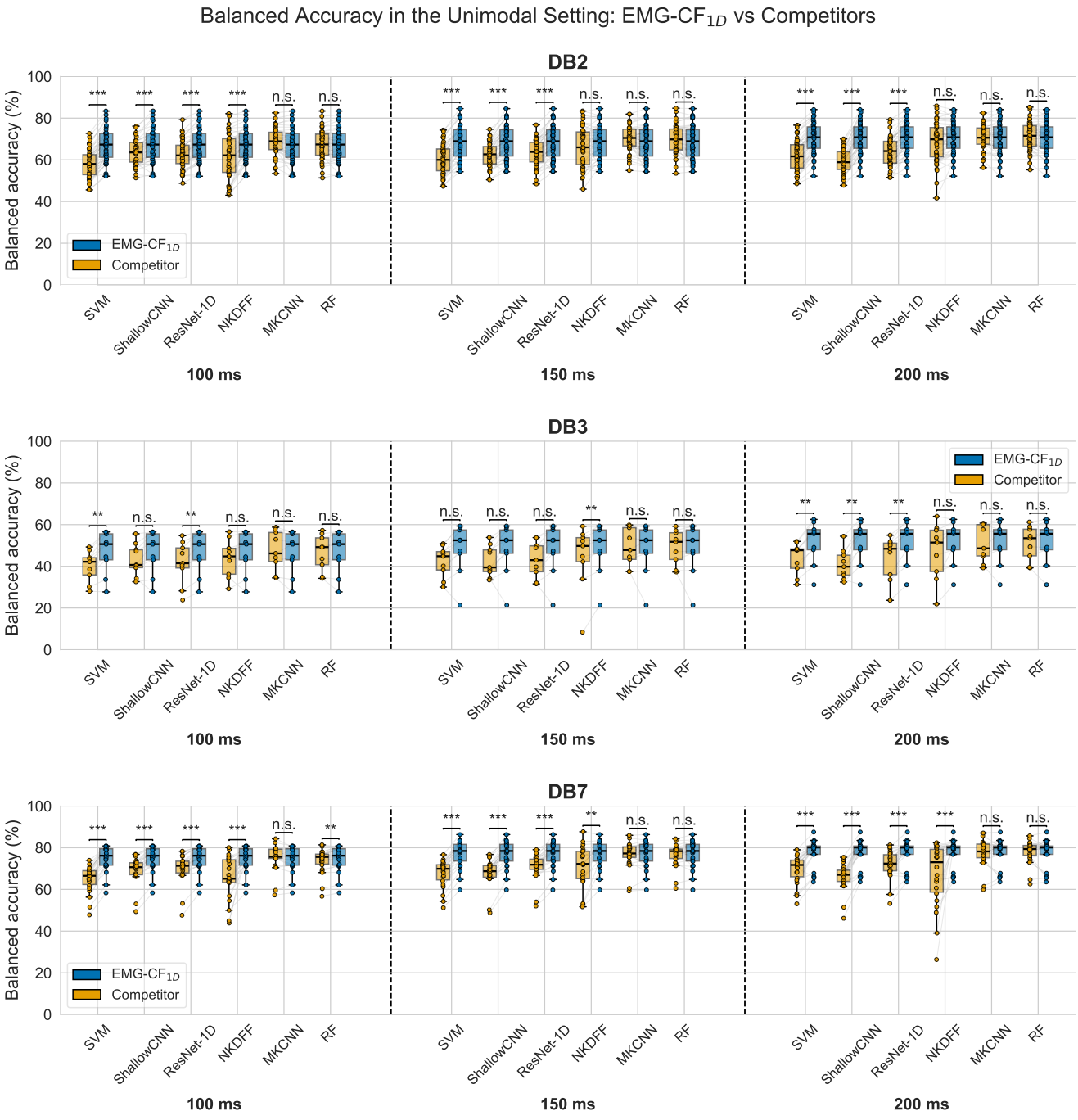

Figure 3: sEMG-only paired comparisons between EMG-CF_{1D} and the competing models for NinaPro DB2, DB3, and DB7 across 100, 150, and 200 ms windows. Boxplots summarize the subject-level balanced accuracies; points represent individual subjects, and gray lines connect paired observations. Asterisks indicate one-sided paired Wilcoxon signed-rank tests adjusted across the 108 unimodal comparisons using the Benjamini–Hochberg FDR procedure: $^{***}p_{\mathrm{FDR}} < 0.001$ and $^{**}p_{\mathrm{FDR}} < 0.01$. Comparisons with $p_{\mathrm{FDR}} \geq 0.01$ are denoted n.s.

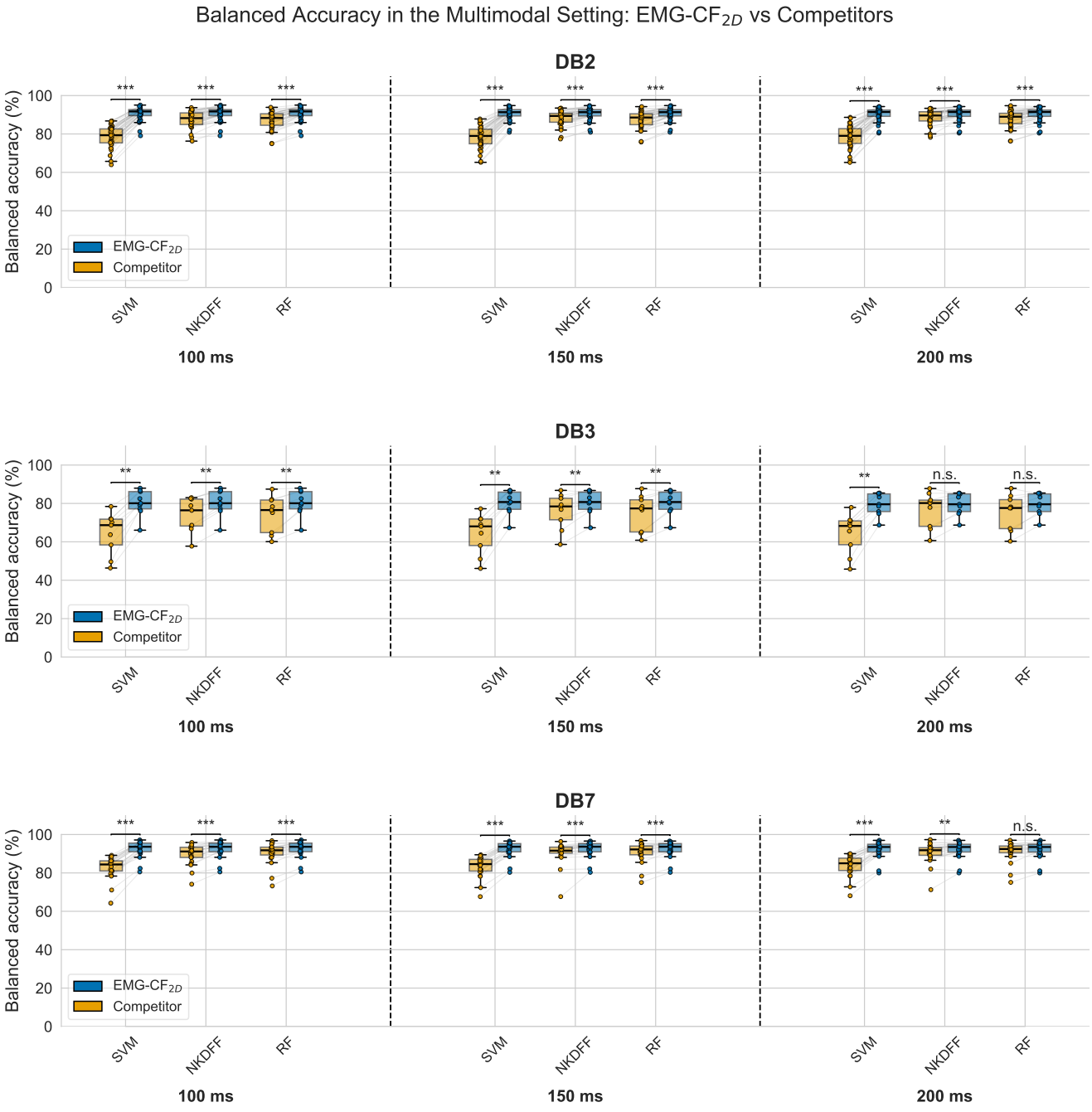

Figure 4: sEMG + ACC paired comparisons between EMG-CF_{2D} and the competing multimodal models for NinaPro DB2, DB3, and DB7 across 100, 150, and 200 ms windows. Boxplots summarize the subject-level balanced accuracies; points represent individual subjects, and gray lines connect paired observations. Asterisks indicate one-sided paired Wilcoxon signed-rank tests adjusted across the 54 multimodal comparisons using the Benjamini–Hochberg FDR procedure: $^{***}p_{\mathrm{FDR}} < 0.001$ and $^{**}p_{\mathrm{FDR}} < 0.01$. Comparisons with $p_{\mathrm{FDR}} \geq 0.01$ are denoted n.s.

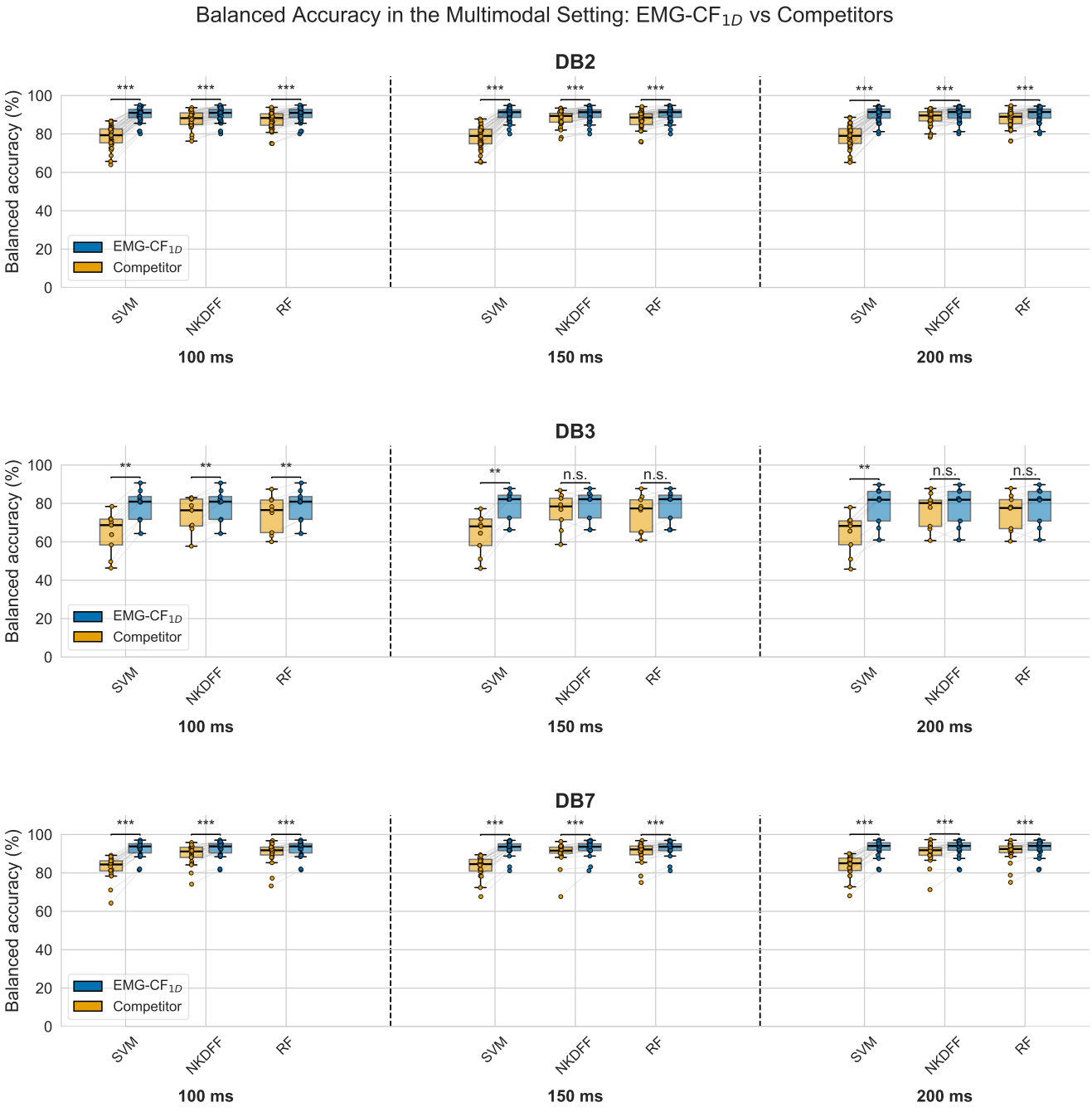

Figure 5: sEMG + ACC paired comparisons between EMG-CF$_{1D}$ and the competing multimodal models for NinaPro DB2, DB3, and DB7 across 100, 150, and 200 ms windows. Boxplots summarize the subject-level balanced accuracies; points represent individual subjects, and gray lines connect paired observations. Asterisks indicate one-sided paired Wilcoxon signed-rank tests adjusted across the 54 multimodal comparisons using the Benjamini–Hochberg FDR procedure: $^{***}p_{\mathrm{FDR}} < 0.001$ and $^{**}p_{\mathrm{FDR}} < 0.01$. Comparisons with $p_{\mathrm{FDR}} \geq 0.01$ are denoted n.s.

A.4. Summary results with other metrics

Table 7 and Table 6 compare EMG-CrossFormer with the other models using additional metrics, namely Cohen's kappa and the F1-score. Regardless of the metric used, EMG-CrossFormer maintains superior performance, demonstrating stronger decoding capabilities than competing models.

Table 6: F1-score across models, datasets, window lengths, and number of modalities

Model	Modality	DB2 100ms	DB2 150ms	DB2 200ms	DB3 100ms	DB3 150ms	DB3 200ms	DB7 100ms	DB7 150ms	DB7 200ms
SVM	sEMG	0.60±0.07	0.62±0.07	0.64±0.07	0.41±0.07	0.43±0.07	0.44±0.07	0.65±0.06	0.68±0.06	0.70±0.07
	sEMG + ACC	0.80±0.05	0.80±0.05	0.81±0.05	0.66±0.10	0.66±0.09	0.66±0.09	0.83±0.06	0.84±0.05	0.84±0.05
Random Forest	sEMG	0.68±0.07	0.70±0.07	0.72±0.07	0.46±0.08	0.49±0.08	0.50±0.08	0.74±0.06	0.76±0.06	0.78±0.06
	sEMG + ACC	0.88±0.04	0.88±0.04	0.88±0.04	0.75±0.09	0.76±0.09	0.76±0.09	0.90±0.06	0.91±0.05	0.91±0.05
ShallowCNN	sEMG	0.63±0.06	0.62±0.06	0.59±0.06	0.42±0.07	0.42±0.07	0.40±0.06	0.69±0.07	0.67±0.07	0.66±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
ResNet-1D	sEMG	0.62±0.07	0.63±0.06	0.64±0.07	0.41±0.10	0.42±0.08	0.43±0.10	0.70±0.07	0.70±0.07	0.72±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
MKCNN	sEMG	0.68±0.07	0.70±0.06	0.70±0.06	0.47±0.09	0.48±0.08	0.50±0.08	0.75±0.06	0.76±0.06	0.77±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
NKDFF	sEMG	0.62±0.11	0.66±0.10	0.69±0.10	0.43±0.08	0.44±0.15	0.46±0.13	0.65±0.11	0.71±0.10	0.66±0.16
	sEMG + ACC	0.89±0.04	0.89±0.04	0.89±0.04	0.75±0.08	0.77±0.08	0.77±0.08	0.90±0.05	0.90±0.06	0.91±0.06
EMG-CF$_{1D}$	sEMG	0.67±0.08	0.68±0.08	0.70±0.08	0.46±0.10	0.47±0.11	0.50±0.10	0.75±0.06	0.77±0.07	0.77±0.07
	sEMG + ACC	0.90±0.04	0.90±0.03	0.90±0.04	0.80±0.07	0.79±0.07	0.79±0.09	**0.92±0.04**	**0.93±0.04**	**0.93±0.04**
EMG-CF$_{2D}$	sEMG	**0.70±0.07**	**0.71±0.06**	**0.72±0.06**	**0.48±0.10**	**0.50±0.10**	**0.52±0.08**	**0.77±0.06**	**0.78±0.06**	**0.79±0.07**
	sEMG + ACC	**0.91±0.03**	**0.91±0.03**	**0.90±0.03**	**0.81±0.06**	**0.81±0.05**	**0.80±0.05**	**0.92±0.04**	0.92±0.04	0.92±0.04

Table 7: Cohen's Kappa across models, datasets, window lengths, and number of modalities

Model	Modality	DB2 100ms	DB2 150ms	DB2 200ms	DB3 100ms	DB3 150ms	DB3 200ms	DB7 100ms	DB7 150ms	DB7 200ms
SVM	sEMG	0.55±0.07	0.57±0.08	0.59±0.08	0.37±0.07	0.39±0.07	0.40±0.07	0.63±0.07	0.66±0.07	0.68±0.08
	sEMG + ACC	0.77±0.06	0.76±0.06	0.77±0.06	0.63±0.10	0.62±0.10	0.62±0.10	0.82±0.06	0.82±0.06	0.83±0.06
Random Forest	sEMG	0.65±0.08	0.68±0.07	**0.70±0.07**	0.44±0.08	0.47±0.08	0.48±0.07	0.73±0.06	0.75±0.06	0.77±0.06
	sEMG + ACC	0.86±0.05	0.86±0.05	0.87±0.05	0.73±0.09	0.74±0.09	0.75±0.09	0.90±0.06	0.90±0.06	0.91±0.06
ShallowCNN	sEMG	0.61±0.06	0.60±0.06	0.57±0.06	0.40±0.07	0.40±0.07	0.38±0.07	0.68±0.07	0.66±0.07	0.65±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
ResNet-1D	sEMG	0.60±0.07	0.61±0.07	0.62±0.07	0.39±0.09	0.41±0.08	0.42±0.09	0.69±0.08	0.69±0.07	0.70±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
MKCNN	sEMG	0.66±0.07	0.68±0.07	0.69±0.06	0.45±0.09	0.46±0.08	0.48±0.08	0.74±0.07	0.75±0.07	0.76±0.07
	sEMG + ACC	—	—	—	—	—	—	—	—	—
NKDFF	sEMG	0.59±0.11	0.63±0.10	0.66±0.10	0.40±0.09	0.41±0.14	0.44±0.13	0.63±0.11	0.69±0.11	0.65±0.16
	sEMG + ACC	0.86±0.04	0.87±0.04	0.87±0.04	0.72±0.08	0.75±0.09	0.75±0.09	0.89±0.06	0.89±0.07	0.90±0.06
EMG-CF$_{1D}$	sEMG	0.64±0.08	0.66±0.08	0.68±0.08	0.44±0.09	0.45±0.11	0.49±0.09	0.73±0.07	0.76±0.07	0.76±0.07
	sEMG + ACC	0.89±0.04	0.90±0.04	0.89±0.04	0.78±0.08	0.78±0.08	0.78±0.10	**0.92±0.04**	**0.92±0.04**	**0.92±0.05**
EMG-CF$_{2D}$	sEMG	**0.68±0.07**	**0.69±0.07**	**0.70±0.07**	**0.46±0.09**	**0.48±0.10**	**0.50±0.08**	**0.76±0.07**	**0.77±0.07**	**0.78±0.07**
	sEMG + ACC	**0.90±0.04**	**0.90±0.03**	**0.90±0.04**	**0.79±0.07**	**0.80±0.06**	**0.79±0.06**	0.92±0.05	**0.92±0.04**	**0.92±0.05**