2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

A Novel Framework for sEMG Gesture Recognition Based on Soft Prompt Learning

Dingchi Sun[1] and Junjian Ren[1]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, P.~R.~China

Abstract. Surface electromyography (sEMG) signals hold considerable promise for predicting human motion prior to its actual execution. However, a major challenge in sEMG-based intention recognition lies in the severe noise interference and high inter-subject variability inherent in traditional myoelectric time-series signals. These issues hinder accurate alignment with corresponding actions and constrain model learning capacity. To address these challenges, this study proposes a dual-modal contrastive learning framework based on Contrastive Language-Audio Pretraining (CLAP). By introducing textual prompts as auxiliary guidance for interpreting sEMG signals, the proposed method enhances recognition accuracy while reducing redundant training. In addition, a k-layer hierarchical processing algorithm is developed to expand the training dataset to a quadratic scale of its original size, thereby mitigating the problem of limited data availability and facilitating integrated prediction. The proposed approach is evaluated on public benchmark datasets, including Ninapro DB1, DB2, DB5, and CapgMyo. Experimental results show that the model outperforms state-of-the-art (SOTA) methods by 2-3%.

Keywords: Surface Electromyograph, Gesture Recognition, Segmentation Parameters, Multimodal learning, Contrastive Learning.

1 Introduction

Surface electromyography (sEMG) signals are collected using electrodes attached to the skin's Surface [1]. Due to their advantages of motion anticipation and non-invasiveness, sEMG-based intent recognition typically involves three key steps: data preprocessing, feature extraction, and model training [2]. This technique has been widely applied in medical diagnostics, exoskeleton control, and human-computer interaction.

sEMG signals are highly susceptible to noise interference, such as power line interference, motion artifacts, and baseline drift. Preprocessing aims to enhance signal quality and standardize data. This includes filtering and denoising, where band-pass filtering is used to remove high-frequency noise and low-frequency baseline drift, while notch filters eliminate 50/60 Hz power line interference [3]. Signal segmentation is performed using a sliding window to divide continuous signals into short segments, balancing real-time processing and information integrity [4]. Standardization methods,

such as Z-score normalization, eliminate individual differences in muscle strength and the influence of equipment gain [5].

Feature extraction involves identifying key characteristics from the preprocessed signal that represent muscle activity patterns while balancing computational efficiency and classification performance. These features can be categorized as follows: time-domain features, such as Mean Absolute Value, Root Mean Square, and Zero Crossing Rate, which reflect signal amplitude and complexity [5,6] , frequency-domain features, where Fourier transform extracts parameters like Median Frequency and Mean Power Frequency to indicate muscle fatigue states; time-frequency features, such as Wavelet Transform or Short-time Fourier Transform, which capture the dynamic changes in the signal [7], and higher-order features, such as entropy measures (e.g., sample entropy, approximate entropy), which further enhance the representation of muscle activity patterns [8,9] .

Based on the extracted features, intent recognition models are constructed. Traditional machine learning methods, such as support vector machines, random forests, and linear discriminant analysis, rely on handcrafted feature engineering and are well-suited for small-sample scenarios [10,11] . Deep learning approaches, such as convolutional neural networks (CNNs), automatically learn spatial and temporal features, making them effective for multi-channel sEMG signals. Recurrent neural networks (RNN) and long short-term memory networks (LSTM) capture the temporal dependencies in signals, improving dynamic motion recognition accuracy [12]. Hybrid models, such as CNN-LSTM architectures, combine spatial and temporal features, while graph neural networks model muscle coordination relationships [13,14] .

Despite significant advancements, challenges remain in sEMG-based gesture recognition, including individual signal variability, dynamic environmental interference, limited labeled data, and insufficient model generalization [15]. Research efforts have explored solutions such as transfer learning, data augmentation using generative adversarial networks to synthesize realistic data, and multimodal fusion [16,17] . The ultimate goal is to transition from controlled laboratory environments to practical applications with high robustness and low latency, providing critical technological support for intelligent rehabilitation and human-computer interaction.

Our starting point is that the existence of domain differences leads to individual model variations [18]. However, the basic movements of each subject can serve as a text prompt to enhance gesture prediction accuracy. With the development of large time-series models, which demonstrate strong feature extraction capabilities [19-21]. One of the key research directions in sEMG-based gesture recognition is how to achieve high accuracy based on individual subject data. The contributions of this paper are as follows:

1. To address individual difference in sEMG signals, We introduce a multimodal approach based on CLAP, and design a novel learning framework for gesture recognition. This framework combines the feature extraction capabilities of large time-series models and text encoders, which serve as a text prompt-based guidance.

2. To address data deficiencies in the sEMG signals, we design a new processing algorithm called k-layer window-based algorithm. This algorithm can enhance data diversity during pre-processing through data augmentation, and also improve prediction accuracy during post-processing via ensemble learning.
3. We conduct experimental validation on multiple datasets, demonstrating the effectiveness of our model and algorithm. Our model achieves 1-2% higher accuracy than previous approaches, while our algorithm, as a plug-and-play module, significantly enhances classifier performance.

2 Related Work

2.1 sEMG-based Gesture Recognition

The key to sEMG gesture recognition lies in extracting effective features from raw EMG signals. Existing research primarily focuses on time-domain, frequency-domain, and time-frequency domain feature extraction [5-9]. Time-domain features offer high computational efficiency, making them suitable for real-time applications. Frequency-domain features reflect the frequency characteristics of muscle contractions. Regarding classification algorithms, traditional machine learning methods such as support vector machines, random forests, and k-nearest neighbors have been widely used [10,11] . However, these methods rely on manually designed features and have limitations when dealing with noisy sEMG signals.

In recent years, deep learning approaches, including CNNs, LSTM networks, and transformers, have gradually become mainstream. Hu et al. propose an attention-based hybrid CNN-RNN architecture with a new sEMG image representation method[22]. Ma et al. proposed a short connected autoencoder long short-term memory based simultaneous and proportional scheme [14]. Zhu et al. proposes an improved PCA-based CNN-LSTM model for accurate lower limb activity prediction from sEMG signals, enabling real-time myoelectric control of exoskeletons [13]. In conclusion, CNNs effectively extract spatial features, while LSTMs are well-suited for handling sequential information, enhancing dynamic gesture recognition performance. Moreover, hybrid models such as CNN-LSTM combine spatial and temporal information, achieving high recognition accuracy across multiple datasets[10-12].

2.2 Times Series Model and Prompt Finetuning

As a type of time-series signal, sEMG differs from other time signals in its strong nonlinearity, non-stationarity, low signal-to-noise ratio, and significant susceptibility to factors such as muscle fatigue and electrode displacement. However, the recent development of advanced time-series models has demonstrated strong feature extraction capabilities and generalization performance, offering new perspectives for addressing the challenges in sEMG signal processing [21,23,24]. Thus, how to effectively adapt large pre-trained models for downstream tasks has become a key research focus.

Recent studies explore prompt-based fine-tuning approaches without full retraining.

2.3 Motivation

Due to the individual variability of sEMG signals and the inevitability of concept drift, contrastive learning can adapt to various changes while optimizing data and knowledge accumulation. However, the inherent complexity of contrastive learning algorithms, the influence of data noise, and the limited volume of sEMG data pose major challenges to their application in sEMG-based gesture recognition.

This paper proposes a constractiv learning-based sEMG gesture recognition framework inspired by the CLAP architecture. The framework adopts a multimodal contrastive learning structure that integrates a temporal large model with a text encoder, thereby reducing the burden of purely contrastive learning while enhancing its applicability to sEMG scenarios. Additionally, a K-layer window-based internal algorithm is designed to increase data diversity and volume. Our method is shown in **Fig. 1.**

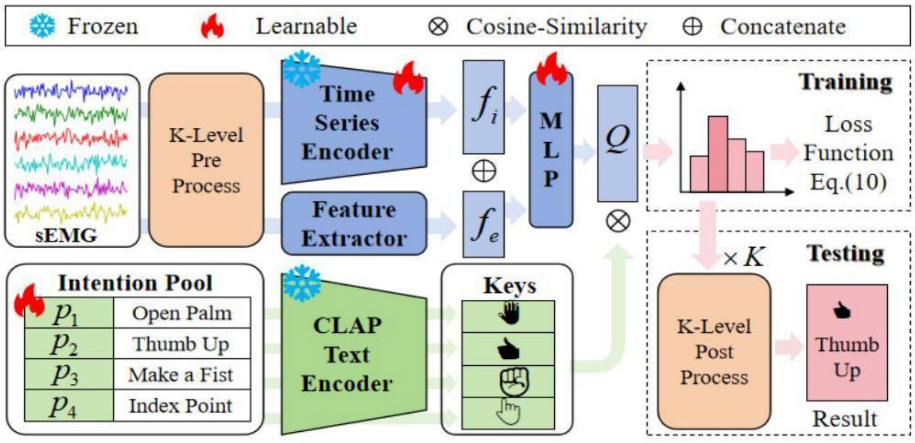

Fig. 1. Overview of the Proposed Framework: Fine-tuning a temporal large model to leverage its generalization capability, the framework integrates both explicit and implicit features. A contrastive learning strategy is employed to extract semanticrepresentations from label-referenced textual prompts, while soft prompt learning is utilized to align the feature distributions of labels and sEMG signals in a shared representation space, thereby enabling accurate intention recognition.

3 Methodology

3.1 Data Processing

Before training the model, the electromyography (EMG) data $x_i \in \mathbb{R}^{l \times c}$ undergoes preprocess. Since the effective frequency range of EMG signals is $20 - 500$ Hz , we follow the conventional approach of applying a 4th-order Butterworth filter and a 1st-order low-pass filter. Typically, min-max normalization is used, but some studies have demonstrated the effectiveness of μ -law normalization, which is a nonlinear method commonly applied in audio signal processing.

Specifically, it is defined as:

$$X_{\text{norm}} = \frac{\ln\left(1+\mu\cdot\frac{|X|}{X_{\max}}\right)}{\ln(1+\mu)}\cdot \text{sign}(X) \tag{1}$$

where μ is the compression parameter (default value 255), $X_{\max}$ represents the maximum absolute value of the input signal, and $\text{sign}(X)$ retains the sign of the original data.

Studies have shown that longer window lengths provide richer information, improving recognition accuracy. However, excessively long windows may affect the responsiveness during prosthetic limb usage. Therefore, we adopt the windowing settings used in previous studies.

3.2 Contrastive Learning Based on Prompts

Suppose there are T subjects, and their sEMG data sample spaces are $D = \{D^1, D^2, \dots, D^T\}$. For the t-th subject, the data is $D^t = \{X_t, Y_t\}$, where $X_t = \{x_i\}_{i=1}^{n_t}$ and $Y_t = \{y_i\}_{i=1}^{n_t}$. Let the text prompt be $[p_1, p_2, \dots, p_n]$. The prediction model is denoted as $f(\cdot)$, and its output can be guided using the text prompt. In sEMG based intent recognition, the goal of contrastive prompt learning is to learn a model based on text prompts for subjects to make similar sample representations closer. By incorporating text prompts as a soft prompt method, a specific topic model with strong adaptability can be trained with minimal increase in model parameters. The formal statement is as follows:

$$\min_f E_{(x,y)\sim D}[\mathcal{L}(f(x;p),y)] \tag{2}$$

where $\mathcal{L}$ represents the loss function, such as cross-entropy loss. In CLIP or CLAP based models, contrastive loss is typically used.

3.3 Intention Recognition Based on CLAP

Let the temporal prediction model be E_{time}, the text encoder be E_{txt}, and K_{pre} be the hierarchical preprocessing, which will be detailed in the next section. For a given sample from a subject, i.e., an sEMG window signal $x_i \in \mathbb{R}^{l\times c}$, the intention label is $y_i \in Y_b$. Under the CLAP framework, similar to Section 3.1, we also introduce an intention text Prompt Pool $[p_1, p_2, \dots, p_b]$, but this pool is directly related to the intention labels. During training, the data is first processed hierarchically as follows:

$$\{x_i^k\}_{k=1}^{\widehat{K}} = K_{\text{pre}}(x_i) = \{x_i^1, x_i^2, \dots, x_i^K\} \tag{3}$$

where $\widehat{K} = \frac{K(K+1)}{2}$ (see Section 3.4). Each x_i^n is then processed by the two encoders. The time model is as follows:

$$f_e = E_{\text{time}}(x_i^n; P_t) \tag{4}$$

Here, $f_e \in \mathbb{R}^{C_{\text{time}}}$ represents explicit features. To achieve domain adaptation and feature fusion, after the model extracts implicit features, we introduce explicit features using a traditional feature extractor FT :

$$f_i = FT\left(x_i^k\right) \tag{5}$$

$$f_{\text{fuse}} = \text{Concat}(f_e, f_i) \tag{6}$$

where $f_{\text{fuse}} \in \mathbb{R}^{C_{\text{time}} + C_{ft}}$. To achieve dimension alignment and further fuse explicit and implicit features, we use an MLP fully connected layer:

$$Q = MLP(f_{\text{fuse}}) \tag{7}$$

such that $Q \in \mathbb{R}^b$. Simultaneously, the intention label y_i is used to guide the training of the text encoder prompt:

$$K = E_{txt}\left(P_{y_i}^t; [\text{INI\_NAME}]\right) \tag{8}$$

where $K_c \in \mathbb{R}^b$. Now, Q and K can be compared using cosine similarity: The cosine similarity is computed as:

$$\langle Q, K_c \rangle = \frac{Q \cdot K_c}{\|Q\| \cdot \|K_c\|} \tag{9}$$

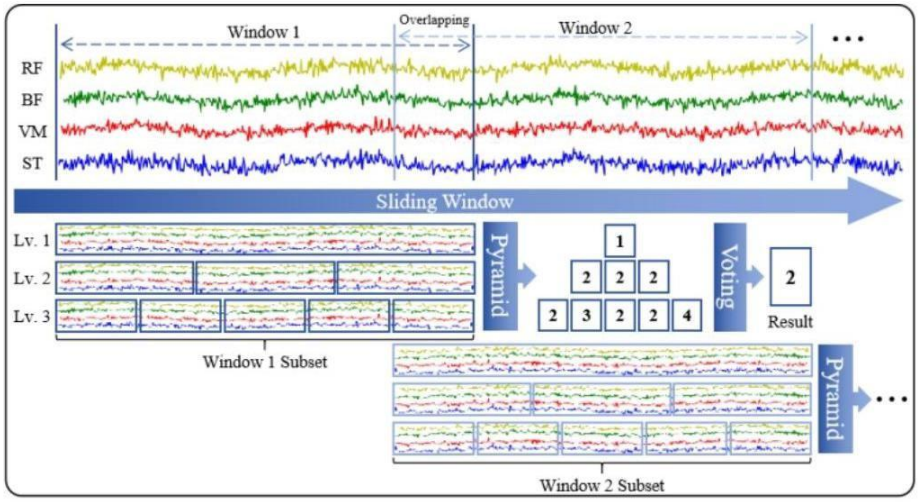

Fig. 2. Schematic diagram of K-layer Window-based Algorithm

Our loss function is designed as:

$$\mathcal{L}_{CE} = -\frac{1}{|D_t|}\sum_{i=1}^{|D_t|} \log p(y_i = c \mid x_i) \tag{10}$$

where the probability is calculated by:

$$p(y_i = c \mid x_i) = \frac{\exp(\langle Q_i, K_c \rangle / \tau)}{\sum_{c' \in y_t} \exp(\langle Q_i, K_{c'} \rangle / \tau)} \tag{11}$$

where τ is the temperature parameter that controls the model output.

3.4 K-layer Window-based Algorithm

The internal hierarchical voting algorithm consists of two parts: the preprocessing stage K_{pre} for training the model and the post-processing stage K_{post} for aggregating results. For the k -th layer, there exist K sub-windows to augment training data.

$$\{x_i^k\}_{k=1}^{\hat{K}} = K_{\text{pre}}(x_i) = \{x_i^1, x_i^2, \ldots, x_i^{\hat{K}}\} \tag{12}$$

Specifically, for data of length 1, each k -layer can augment k pieces of data. The step length is determined by an overlap ratio of approximately 20% , given by:

$$l_k = \lfloor L \cdot (1/K) \times 1.2 \rfloor \tag{13}$$

Algorithm 1 K-layer Window-based Algorithm

Input: Sample data $x_i \in \mathbb{R}^{l \times c}$; number of layers K ; model M ; current stage S
Output: Trained model M or predicted result $\hat{y}_i$
Initialize $\{x_i^k\}_{k=1}^{\hat{K}} = \{x_i\}$ Obtain initial window length l_1
for $k = 2$ to K **do**
  Compute window length: $l_k = \left\lfloor \frac{L}{K} \times 1.2 \right\rfloor$ Set stride: $s = 0.2 \times l$
  Slice x_i into k samples:
    $\text{Slice}(k) = \{x_i[0: l_k], x_i[l_k - s: 2l_k - s], \ldots\}$
    Append slices: $\{x_i^k\}_{k=1}^{\hat{K}} \leftarrow \{x_i^k\}_{k=1}^{\hat{K}} \cup \text{Slice}(k)$
end
if *S is training* **then**
  Train model M until convergence:
    $M \leftarrow \text{Model.training}(M, \{x_i^k\})$
  return Trained model M
end
else
  for $k = 1$ *to* K **do**
    $\hat{y}_i^k = \text{Model.predict}(M, x_i^k)$
  end
  Aggregate predictions:
    $\hat{y}_i = \text{Voting}\left(\{\hat{y}_i^k\}_{k=1}^{\hat{K}}\right)$
  return Prediction result $\hat{y}_i$
end

This step-length-adjusted data slicing operation is applied to each channel of x . The schematic diagram of our method is shown in **Fig. 2** The specific implementation of the algorithm is shown in Algorithm 1.

Through operations, each window can be divided into multiple sub-windows. In actual training, we use k^2 sub-windows for training, which allows the training set to be increased by a factor of R . If only the preprocessing stage K_{pre} is used to train the model, the algorithm's complexity remains $O(1)$ during testing. However, if post-processing K_{post} is applied to aggregate the results, the algorithm's time complexity becomes $O(K^2)$.

Deep learning models generally require inputs of the same size. To achieve this, we employ Center Padding, defined as:

$$x_i^k = \text{Padding}\left(x_i^k; l\right) \tag{14}$$

Here, padding preserves the existing data at the center by filling zeros before and after the data to reach the target length.

When the chosen number of layers reduces the window length to half of the original length, i.e., $l_k < 0.5 \times l$, it adversely affects the training process. This is analogous to introducing a Dropout layer at the input level with a rate of 0.5, whereas the typical value ranges from 0 to 0.5 . Therefore, we set the maximum number of layers K to 5, as demonstrated in the subsequent ablation experiments.

4 Experiments and Results Analysis

4.1 Experimental Setup

sEMG datasets settings

We validate our model and algorithm using commonly used sEMG datasets: Ninapro DB1, DB2, DB5 and Capmygo.

Ninapro dataset [25] is the largest data collection effort for sEMG signals, comprising 10 extensive databases collected from both amputees and intact subjects using various sensors. In our experiments, we utilized NinaPro DB1, DB2, and DB5 for validation, with sampling rates of 100 Hz, 2000 Hz , and 200 Hz , respectively.

Capmygo DB-a [26] consisted of recordings of 8 finger gestures, using the 128 channels HD-sEMG signals recorded by our non-invasive wearable device with sampling rates of 1000Hz .

Their specific designs are shown in **Table 1.** Dataset settings:

Table 1. Dataset settings

Dataset	Subjects	Rounds	Train	Test	Channels
DB1	27	10	1,2,5,7,10	3, 6, 9	10
DB2	40	6	1,3,5,6	2,4	8
DB5	10	6	1,3,5,6	2,4	8
Mygo	18	10	1,2,5,7,10	3, 6, 9	128

We conducted experimental operations on the sEMG signals of each subject in all datasets and report the final average accuracy and standard deviation for different models.

Pretrained model for CLAP

Table 2. Average Accuracy (%) of Different Models on Upper Limb Datasets

Model	Year	Upper	
		DB1-100	DB5-200
CNN+LSTM	2018	81.23	73.23
LightTS	2024	80.67	69.34
DLinear	2024	74.05	67.45
iTransformer	2024	83.47	72.31
TDCT	2024	85.15	72.23
Ours	2025	**87.78**	**74.83**

For the time series model, in order to achieve good implicit feature extraction capability while implementing Prompt, we adopt the Transformer-based large model $\text{MOMENT}_{\text{small}}$ version. This model has a high feature extraction capability and has been trained on certain sEMG datasets, demonstrating strong zero-shot ability. We set the prompt length for each subject in the Subject Pool to 1 and use a Prefix-based approach for guidance.

For the text model, we use the text generator of the CLAP model, namely BERT, specifically the BERT base uncased version implemented by Hugging-Face. For computational efficiency, we limit the maximum text sequence length to 100 characters. The [CLS] token from the last layer of BERT is used as the text embedding.

Comparison Methods

Regarding model comparisons, to verify the intent recognition capability of various time series models on sEMG data, we compare results using CNN+LSTM, LightTS, DLinear, iTransformer, and TDCT. Additionally, to validate the effectiveness of the K-layer window-based algorithm, we introduce traditional classifiers such as SVM, RF, and KNN for comparison in explicit feature classification.

Explicit Feature Selection

Time-domain feature set: Mean Absolute Value(MAV), Root Mean Square(RMS), Waveform Length(WL).

$$\text{MAV} = \frac{1}{L}\sum_{i=1}^{L}|x_i| \tag{15}$$

$$\text{RMS} = \sqrt{\frac{1}{L}\sum_{i=1}^{L}x_i^2} \tag{16}$$

$$\text{WL} = \sum_{i=2}^{L}|x_i - x_{i-1}| \tag{17}$$

where x_i denotes the sEMG data with a window length of L .

Parameter settings for K

This algorithm does not introduce additional parameters. If only hierarchical partitioning is performed, the training data increases, leading to longer model training time, while the inference speed remains unchanged. However, if hierarchical voting is applied, the number of inferences for a single window will also increase, making it unsuitable for practical applications. Therefore, considering model inference speed, we only investigate the results for $K = 1,2,3,4,5$.

Training Settings and Parameters

We conducted our experiments using four GTX 4090 GPUs. For traditional classifiers, we used t he default parameters unless otherwise specified. During model training, the batch size was set to 256, and the model was trained for 30 epochs.

During training, only the Intention Pool, the MLP layer within MOMENT, and the fused MLP layer were updated, while all other components remained frozen. The number of trainable parameters is summarized as follows:

Table 3. Effect of Module Combination on Model Performance (%)

Model Description	Upper Limb Datasets	
	DB1-100	DB5-200
Baseline (Only E_{time})	83.89	72.04
w E_{text}	85.75	73.47
w F_E	85.32	73.00
w $E_{\text{text}} + F_E$	86.13	73.91
w $E_{\text{text}} + K$ -level	86.84	74.00
Full Model	**87.78**	**74.83**

Note: "w" denotes that the corresponding module is included.

4.2 Result

According to the above experimental results in **Table 2.** Average Accuracy (%) of Different Models on Upper Limb Datasets, the intention recognition model proposed in this study demonstrates outstanding performance across different datasets, achieving the highest accuracy on all four datasets used. Specifically, on the upper limb dataset DB1-100, the proposed model reaches an average accuracy of 87.78%, and on the dataset collected in this study, it achieves an average accuracy of 93.03%, significantly outperforming other comparison models. This indicates that the approach of fine-tuning the temporal large model and integrating explicit and implicit features offers a clear advantage.

5 Ablation Study

5.1 General Ablation

We conducted ablation studies on the components of the model. In **Table 3.** Effect of Module Combination on Model Performance (%), the results show that each component contributes to the optimization of the model, leading to better performance compared to the baseline.

5.2 K-layer Window-based Algorithm Ablation

Different K **of** K **-layer Window-based Algorithm**

As shown in the **Fig. 3**, we present the accuracy variations of traditional classifiers and deep learning classifiers as K changes, exploring the impact of different K values across various datasets. The changes after hierarchical voting are illustrated.

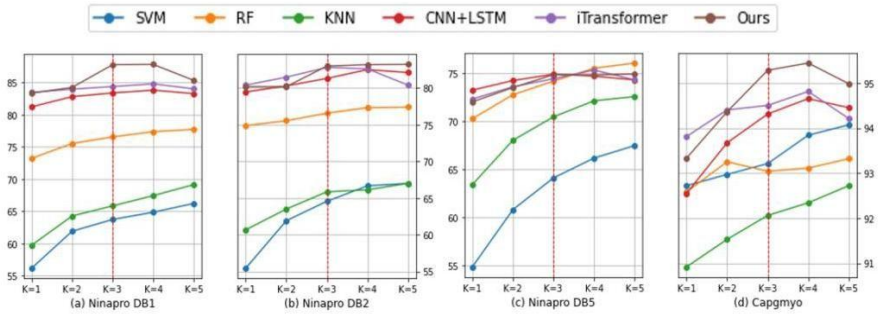

Fig. 3. Results of several methods at different K values

It can be observed that for feature-based traditional classifiers, increasing K significantly improves accuracy, especially for SVM, which can achieve up to a 10% improvement. However, for deep learning-based models, classification results generally improve when K is between 1 and 3 . Yet, as K continues to increase, issues related to Padding Pooling, discussed in Section 3, cause the training performance of some classifiers to deteriorate, leading to a decline in accuracy.

Pre and Post Process Ablation of Algorithm

To verify the effectiveness of preprocessing and post-processing, we conducted validation experiments with $K = 3$, and the obtained results are shown in **Table 4.** Comparison of Different Models with and without Preprocessing across Four Datasets (Accuracy %, Standard Deviation).

The above experimental results demonstrate the effectiveness of the hierarchical algorithm. For feature-based classifiers, this algorithm enriches feature diversity, thereby expanding the training sample space. For deep learning-based models, it enhances generalization performance. Based on the algorithm process and experimental results, we

can conclude that even when only preprocessing is applied-by increasing the amount of training data without adding any inference time - accuracy can still be improved.

Table 4. Comparison of Different Models with and without Preprocessing across Four Datasets (Accuracy %, Standard Deviation)

Model	Setting	DB1-100	DB2-200	DB5-200	Capgmyo
SVM	-	56.2 ± 6.3	55.4 ± 6.5	53.6 ± 6.1	92.8 ± 4.2
	Preprocessing	60.0 ± 5.4	60.3 ± 6.0	59.9 ± 5.9	93.4 ± 4.2
	Postprocessing	63.7 ± 5.8	64.6 ± 5.9	64.1 ± 6.0	93.2 ± 3.8
RF	-	73.2 ± 5.0	74.7 ± 5.1	72.5 ± 4.0	92.8 ± 3.6
	Preprocessing	75.4 ± 4.2	76.4 ± 4.6	75.4 ± 4.9	93.4 ± 3.4
	Postprocessing	76.6 ± 4.8	76.6 ± 4.7	76.6 ± 5.0	93.9 ± 3.2
KNN	-	59.8 ± 6.2	60.7 ± 6.6	58.8 ± 6.1	92.5 ± 4.4
	Preprocessing	65.5 ± 6.1	67.3 ± 6.1	66.1 ± 6.2	93.4 ± 3.9
	Postprocessing	67.1 ± 6.3	68.5 ± 6.1	68.8 ± 6.4	93.9 ± 3.5
CNN+LSTM	-	82.7 ± 6.7	81.2 ± 6.8	81.3 ± 6.9	92.3 ± 2.7
	Preprocessing	83.4 ± 5.1	83.1 ± 5.5	83.5 ± 5.4	93.1 ± 2.3
	Postprocessing	84.0 ± 4.8	84.1 ± 5.3	84.9 ± 5.1	94.3 ± 2.1
iTransformer	-	83.9 ± 3.0	82.3 ± 4.4	83.5 ± 4.7	92.9 ± 2.7
	Preprocessing	84.4 ± 3.7	84.2 ± 4.1	84.8 ± 4.2	94.1 ± 2.4
	Postprocessing	85.0 ± 3.4	85.3 ± 4.4	85.4 ± 4.1	94.5 ± 2.1

6 Conclusion

To address key limitations in lower-limb intention recognition using sEMG and FMG signals, this study presents an integrated framework that encompasses data acquisition, parameter optimization, and model design. A multimodal biosig-nal acquisition system was developed to collect synchronized sEMG and FMG data with visual prompts for lower-limb movements, enabling real-time, multichannel recording and facilitating comparative analysis of signal performance. To improve efficiency and interpretability in temporal window parameter selection, a novel method was introduced that identifies optimal combinations of window length and overlap by sampling minimal data and evaluating the trace of between-class and within-class scatter matrices, thereby avoiding extensive classifier retraining. In addition, a K-layer windowing algorithm was designed to enhance data diversity and robustness, with experimental results confirming its effectiveness. Furthermore, to overcome the generalization limitations of conventional intention recognition models, a contrastive learning framework was constructed that leverages temporal large models for implicit feature extraction and integrates explicit features for improved discriminability. Experimental validations demonstrate that the proposed framework significantly enhances recognition accuracy and fine-tuning efficiency, offering a scalable and interpretable solution for real-world applications.

References

1. De Luca, C.J.: The use of surface electromyography in biomechanics. Journal of Applied Biomechanics 13(2), 135–163 (1997).
2. Chowdhury, R.H., Reaz, M.B.I., Ali, M.A.B.M., Bakar, A.A.A., Chellappan, K., Chang, T.G.: Surface electromyography signal processing and classification techniques. Sensors 13(9), 12431–12466 (2013).
3. Clancy, E.A., Morin, E.L., Merletti, R.: Sampling, noise-reduction and amplitude estimation issues in surface electromyography. Journal of Electromyography and Kinesiology 12(1), 1–16 (2002).
4. Mendes Junior, J.J.A., Pontim, C.E., Dias, T.S., Campos, D.P.: How do sEMG segmentation parameters influence pattern recognition process? An approach based on wearable sEMG sensor. Biomedical Signal Processing and Control 81, 104546 (2023).
5. Lehman, G.J., McGill, S.M.: The importance of normalization in the interpretation of surface electromyography: a proof of principle. Journal of Manipulative and Physiological Therapeutics 22(7), 444–446 (1999).
6. Phinyomark, A., Quaine, F., Charbonnier, S., Serviere, C., Tarpin-Bernard, F., Laurillau, Y.: Feature extraction of the first difference of EMG time series for EMG pattern recognition. Computer Methods and Programs in Biomedicine 117(2), 247–256 (2014).
7. Thongpanja, S., Phinyomark, A., Phukpattaranont, P., Limsakul, C.: Mean and median frequency of EMG signal to determine muscle force based on time-dependent power spectrum. Elektronika ir Elektrotechnika 19(3), 51–56 (2013).
8. Shi, J., Cai, Y., Zhu, J., Zhong, J., Wang, F.: sEMG-based hand motion recognition using cumulative residual entropy and extreme learning machine. Medical & Biological Engineering & Computing 51, 417–427 (2013).
9. Shen, C., Pei, Z., Chen, W., Wang, J., Zhang, J., Chen, Z.: Toward generalization of sEMG-based pattern recognition: A novel feature extraction for gesture recognition. IEEE Transactions on Instrumentation and Measurement 71, 1–12 (2022).
10. Wen, T., Zhang, Z., Qiu, M., Zeng, M., Luo, W.: A two-dimensional matrix image based feature extraction method for classification of sEMG: A comparative analysis based on SVM, KNN and RBF-NN. Journal of X-ray Science and Technology 25(2), 287–300 (2017).
11. Cai, S., Chen, Y., Huang, S., Wu, Y., Zheng, H., Li, X., Xie, L.: SVM-based classification of sEMG signals for upper-limb self-rehabilitation training. Frontiers in Neurorobotics 13, 31 (2019).
12. Bittibssi, T.M., Genedy, M.A., Maged, S.A., et al.: sEMG pattern recognition based on recurrent neural network. Biomedical Signal Processing and Control 70, 103048 (2021).
13. Zhu, M., Guan, X., Li, Z., He, L., Wang, Z., Cai, K.: sEMG-based lower limb motion prediction using CNN-LSTM with improved PCA optimization algorithm. Journal of Bionic Engineering 20(2), 612–627 (2023).
14. Ma, C., Lin, C., Samuel, O.W., Xu, L., Li, G.: Continuous estimation of upper limb joint angle from sEMG signals based on SCA-LSTM deep learning approach. Biomedical Signal Processing and Control 61, 102024 (2020).
15. Kumar, D., Ganesh, A.: A critical review on hand gesture recognition using sEMG: Challenges, application, process and techniques. In: *Journal of Physics: Conference Series*, vol. 2327, p. 012075. IOP Publishing (2022).
16. Ao, J., Liang, S., Yan, T., Hou, R., Zheng, Z., Ryu, J.S.: Overcoming the effect of muscle fatigue on gesture recognition based on sEMG via generative adversarial networks. Expert Systems with Applications 238, 122304 (2024).

17. Duan, S., Wu, L., Xue, B., Liu, A., Qian, R., Chen, X.: A hybrid multimodal fusion framework for sEMG-ACC-based hand gesture recognition. IEEE Sensors Journal 23(3), 2773–2782 (2023).
18. Anders, J.P.V., Smith, C.M., Keller, J.L., Hill, E.C., Housh, T.J., Schmidt, R.J., Johnson, G.O.: Inter- and intra-individual differences in EMG and MMG during maximal, bilateral, dynamic leg extensions. Sports 7(7), 175 (2019).
19. Gruver, N., Finzi, M., Qiu, S., Wilson, A.G.: Large language models are zero-shot time series forecasters. Advances in Neural Information Processing Systems 36, 19622–19635 (2023).
20. Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J.Y., Shi, X., Chen, P.Y., Liang, Y., Li, Y.F., Pan, S., et al.: Time-LLM: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728 (2023).
21. Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., Dubrawski, A.: MOMENT: A family of open time-series foundation models. arXiv preprint arXiv:2402.03885 (2024).
22. Hu, Y., Wong, Y., Wei, W., Du, Y., Kankanhalli, M., Geng, W.: A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition. PLOS ONE 13(10), e0206049 (2018).
23. Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: iTransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 (2023).
24. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 11121–11128 (2023).
25. Atzori, M., Gijsberts, A., Heynen, S., Mittaz Hager, A.-G., Deriaz, O., Van Der Smagt, P., Castellini, C., Caputo, B., Müller, H.: Building the Ninapro database: A resource for the biorobotics community. In: 2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp. 1258–1265. IEEE (2012).
26. Dai, Q., Li, X., Geng, W., Jin, W., Liang, X.: Capg-Myo: A muscle-computer interface supporting user-defined gesture recognition. In: Proceedings of the 9th International Conference on Computer and Communications Management, pp. 52–58 (2021).