Article:
Zhang, C., Yu, Z., Wang, X. et al. (3 more authors) (2024) Exploration of deep learning-driven multimodal information fusion frameworks and their application in lower limb motion recognition. Biomedical Signal Processing and Control, 96 (Part B). 106551. ISSN 1746-8094

https://doi.org/10.1016/j.bspc.2024.106551

Exploration of Deep Learning-Driven Multimodal Information Fusion Frameworks and Their Application in Lower Limb Motion Recognition

Changhe Zhang [a], Zidong Yu [a], Xiaoyun Wang [a], Ze-Jian Chen [b,**] (zjchen@hust.edu.cn), Chao Deng [a,*] (dengchao@hust.edu.cn), Sheng Quan Xie [c]

[a] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China.

[b] Department of Rehabilitation Medicine, Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430030, China.

[c] School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT Leeds, U.K.

Abstract:

Research on Lower Limb Motion Recognition (LLMR) based on various wearable sensors has been widely applied in exoskeleton robots, exercise rehabilitation, etc. Typically, employing multimodal information tends to yield higher accuracy and stronger robustness compared to using unimodal information. Due to the inevitable reliance on feature engineering in shallow machine learning-based LLMR methods, this study leverages the powerful non-linear feature mapping capability of deep learning (DL) to construct several end-to-end LLMR frameworks, including: Convolutional Neural Networks (CNNs), CNN-Recurrent Neural Networks (RNNs) and CNN-Graph Neural Networks (GNNs). The effectiveness of the proposed frameworks is verified in distinct tasks, including the recognition of seven types of lower limb motions in healthy subjects and three types of motions in patients with stroke, as well as the phase recognition task during the sit-to-stand (SitTS) process in patients with stroke, achieving the highest mean accuracy of 95.198%, 99.784%, and 99.845%, respectively. Further research and integration of two transfer learning techniques, adaptive Batch Normalization (BN) and model fine-tuning, significantly enhance the applicability of the proposed frameworks in inter-subject prediction. Additionally, systematic analyses are conducted to assess the strengths and weaknesses of different models in terms of recognition performance, complexity, and adaptability to variations in the number of modalities and sensor channels. Experimental results indicate that the proposed frameworks hold promise in providing potential support for the development of human-robot collaborative lower limb exoskeletons or rehabilitation robots.

Keywords: Multimodal information fusion; Lower limb motion recognition; Inter-subject prediction; Deep learning; Transfer learning.

1. Introduction

Exoskeleton is a kind of human-robot cooperative system connected with human body in a wearable way, which has been widely used in motion assistance, exercise rehabilitation and other fields [1-3]. Accurate recognition of the wearer's motion intention is critical for realizing human-robot coordination and active rehabilitation training [4].

In recent years, many studies have emerged on the use of wearable sensors to recognize the motion types of wearers [5]. Depending on the signal type, they can be classified into two

categories: 1) To measure physical changes of human and robot motions by sensors such as goniometer (GON) and accelerometer (ACC) [6-8]. Due to the inherent measurement delay, it is impossible to predict motion before it occurs; 2) To recognize motion patterns by bioelectrical signals, such as Electroencephalogram (EEG) [9], Surface Electromyography (sEMG) [10-12], etc. sEMG signals generated by nerve impulses can be measured before motion occurs and also have the advantage of non-invasive acquisition [13]. However, an issue of concern is that the reliability and robustness of recognition system cannot always be guaranteed by only using single sEMG signals due to factors such as complex physiology and muscle fatigue [14].

Multimodal information fusion is one of the effective solutions for this issue, such as the fusion of EEG and sEMG [14, 15], the fusion of mechanical signals and sEMG [16, 17], etc. The advantages of utilizing multimodal information mainly include: 1) The complementarity of the information provided by different modalities can be used to characterize the subject's motion intention more comprehensively; 2) The system robustness can be greatly enhanced, *i.e.*, reducing the negative impact on recognition accuracy caused by abnormal or lost sensor information from a certain modality. Multimodal information fusion is mainly carried out at the data level, feature level or decision level [15, 18]. The first method usually involves the static weighted fusion of multimodal information, but it is difficult to determine reasonable weight parameters. The third way is the fusion of output decisions from multiple classifiers, and more classifiers mean more computational cost [16]. In contrast, dynamic information fusion at the feature level during the model training process is a more widely applied way [18].

In the past decade, an increasing number of studies have been dedicated to the pattern recognition of sEMG for upper limb movements [19]. However, due to factors such as thick fat and protruding cortex of lower body, the classification of sEMG in lower limbs is more challenging than that in upper limbs [20]. Recently, scholars have explored some LLMR methods using shallow machine learning algorithms [10, 12, 20-23]. Wei et al. combined the single-channel sEMG and Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), etc., to recognize four types of lower limb motions [12]. Zhang et al. proposed a LLMR method using sEMG feature fusion and improved back propagation neural network [22]. However, the aforementioned studies still has some inherent drawbacks. Most notably, operations such as feature extraction and feature selection of sEMG inevitably rely on expert experience, which means that end-to-end LLMR cannot be achieved.

DL, as an important branch of machine learning, has been demonstrated to possess powerful ability to automatically extract deep features, arousing great interest among scholars [11, 24-27]. Vijayvargiya et al. developed a voting-based one-dimensional (1D) CNN model for LLMR in healthy individuals and patients with knee joint injuries [11]. Si et al. constructed the sEMG texture maps, and combined with two-dimensional (2D) CNN to recognize five lower limb motions [24]. Lu et al. proposed two types of RNN models, including CNN-Long–Short Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), for identifying the jump

phases of lower limbs [25]. Wu et al. proposed a gait phase classification method using joint angle signals based on graph convolutional network, a variant model of GNNs [26]. Although these DL-based LLMR studies have greatly eliminated the subjectivity and uncertainty of manual feature extraction and achieved satisfactory recognition performance, there are still some research challenges that cannot be ignored, as described below:

1) For LLMR frameworks constructed with different types of DL models, such as CNNs, RNNs and GNNs and their variants, it is a key to ensure that their feature extractors can fully fuse information from different sensors of different modalities. Traditional studies using simple linear fusion, *i.e.*, feature splicing operation, may lose the importance relationship between sensor channels. Meanwhile, in most LLMR studies, only the recognition accuracy metric is simply evaluated [11, 24-27], lacking multi-dimensional evaluation of DL models, such as the evaluation of time complexity and space complexity metrics that are directly related to the computational efficiency and memory footprint [28]. Additionally, the model adaptability to variations in the number of modalities and sensor channels is also a concern.

2) Nowadays, most researches only focus on the performance of classification models in intra-subject (Intra-S) scenario (a manner of subject-independent training and testing), while their effectiveness in inter-subject (Inter-S) scenario still require further investigation. Since the distribution of training samples is inconsistent with actual test samples, the model pre-trained using data from source domain subjects may be suboptimal or even inadequate when applied to a new target subject. Currently, some transfer learning techniques provide feasible solutions to enhance the domain generalization ability of the model. One is Adaptive BN (AdaBN) [29], whose principle is to use test samples from the target domain to update the mean and variance of the BN layers of the pre-trained model before testing. Since no backpropagation is performed, no additional parameters will be added and the computational cost is almost negligible. Model fine-tuning is another popular technique that aims to transfer knowledge learned from subjects of source domain to the target subject [30, 31]. In practical testing, only limited target domain data and training epochs, *i.e.*, less data acquisition and model training time, are usually required to achieve satisfactory performance on the target subject.

Based on the aforementioned analysis, this paper aims to explore feasible DL-based LLMR frameworks to achieve organic interaction and fusion of multimodal and multi-sensor information, thereby further realizing accurate and efficient LLMR to support the development of human-robot collaborative lower limb exoskeletons. The main contributions are summarized as follows:

1) Four end-to-end LLMR frameworks are presented, including CNN-RNNs, CNN-GNNs and two types of CNNs, and appropriate attention mechanisms are embedded in different frameworks to enhance the feature learning ability of the model. In each framework, existing mainstream network architectures are compared and evaluated from several aspects such as classification performance, complexity and adaptability.

2) In CNN-GNNs framework, two novel graph generation methods, including K-Nearest Neighbor-based Graph (KNNG) and Musculoskeletal Biomechanics-based Graph (MBG), are designed to overcome the defect that multimodal signals have no inherent graph topology. Furthermore, a master nodes-based cross-modal information interaction method is presented.

3) In order to enhance the domain generalization ability of the proposed frameworks in Inter-S prediction, two transfer learning techniques, AdaBN and model fine-tuning, are further integrated and evaluated in terms of recognition performance and time cost.

4) Validation of effectiveness is conducted on datasets of both healthy subjects and patients with stroke, and the effect of using different modal data on recognition performance is analyzed. Additionally, the applicability of the proposed frameworks to the phase recognition task during the SitTS process in patients with stroke is investigated.

The rest of this paper is organized as follows. Section II introduces the basic materials and methods. Section III outlines the proposed frameworks. Section IV presents experimental results and discussion. Section V describes the conclusion and research prospects.

2. Materials and Methods

2.1 Datasets

(1) ENABL3S Dataset

In this case, a publicly available benchmark dataset of bilateral neuromechanical signals called ENABL3S was adopted for experimental validation [32]. It contains the common motion types in community ambulation or daily life and abundant multimodal sensor signals. It was composed of 10 healthy and able-bodied subjects (denoted as S_1-S_{10}) from Northwestern University (7 males and 3 females; 174.8±11.5 cm; 69.4±13.7 kg; 25.5±2.1 years), including signals from mechanical sensors and sEMG electrodes placed in bilateral lower limbs. Fig. 1 shows instrumental setup and the placement of sensors for dataset acquisition.

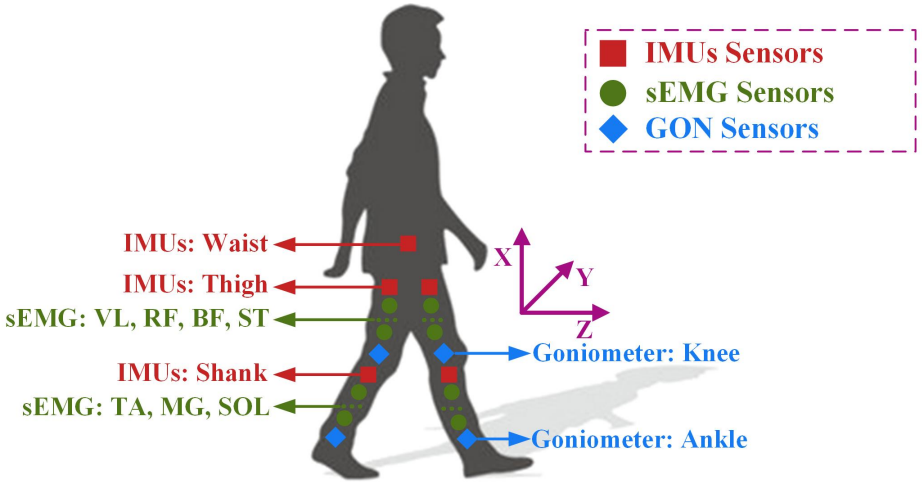


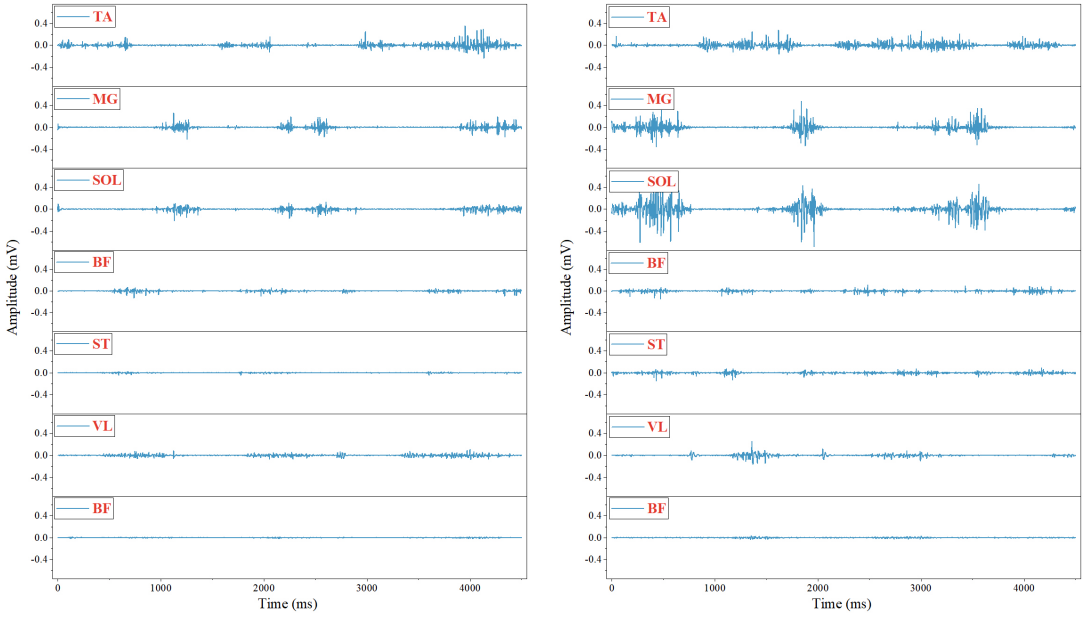
Figure 1. Instrumental setup and the placement of sensors for ENABL3S dataset acquisition.

Fourteen bipolar surface electrodes (DE2.1; Delsys, Boston, MA, USA) were used to measure sEMG signals from the same seven muscles in lower limb, including Tibialis Anterior

(TA), Medial Gastrocnemius (MG), Soleus (SOL), Biceps Femoris (BF), Semitendinosus (ST), Vastus Lateralis (VL), and Rectus Femoris (RF), which were amplified 1000 times, hardware band-pass filtered between 20-450 Hz, and then sampled at 1 kHz. Four 6-DOF (triaxial accelerometer: Ax, Ay and Az; and triaxial gyroscope: Gx, Gy and Gz) Inertial Measurement Units (IMUs) were placed bilaterally on the subjects' thigh and shank, with another attached to the waist, and sampled at 500 Hz (MPU-9250; Invensense, San Jose, CA, USA). In addition, four GONs (SG150; Biometrics Ltd., Newport, UK) were used to collect joint angle signals from bilateral knees and ankles in the sagittal plane, and the sampling frequency was 500 Hz. Data post-processing procedures were as follows: 1) sEMG signals were high-pass filtered at 20 Hz, low-pass filtered at 350 Hz, and notch filtered at 60, 180, and 300 Hz using sixth-order Butterworth filters, respectively; 2) IMU and GON signals were low-pass filtered at 25 and 10 Hz using sixth-order Butterworths, respectively. It should be emphasized that only triaxial ACC signals in each IMUs were used in this study, and both they and GON signals were resampled to 1 kHz to align the time stamps of different sensors when splitting the samples [33].

During an experimental session, each subject was required to repeat two different sequences about 25 times, containing seven activities: Sitting (SIT), Standing (STA), Level Ground Walking (LGW), Ramp Ascending/Descending (RA/RD), and Staircase Ascending/Descending (SA/SD). In odd-numbered trials, the sequence consisted of SIT → STA → LGW → SA → LGW → RD → LGW → STA → SIT, while in even-numbered trials it consisted of SIT → STA → LGW → RA → LGW → SD → LGW → STA → SIT. True motion intention of each subject was labeled using a key fob. In this study, the post-processed data from each subject's first 40 trials were selected as the original experimental dataset.

Fig. 2 shows time-domain waveforms of multimodal signals in bilateral lower limbs acquired by subject S_1 under the motion type of LGW in the first trail, including 14 channels of sEMG signals, 15 channels of ACC signals, and 4 channels of GON signals.



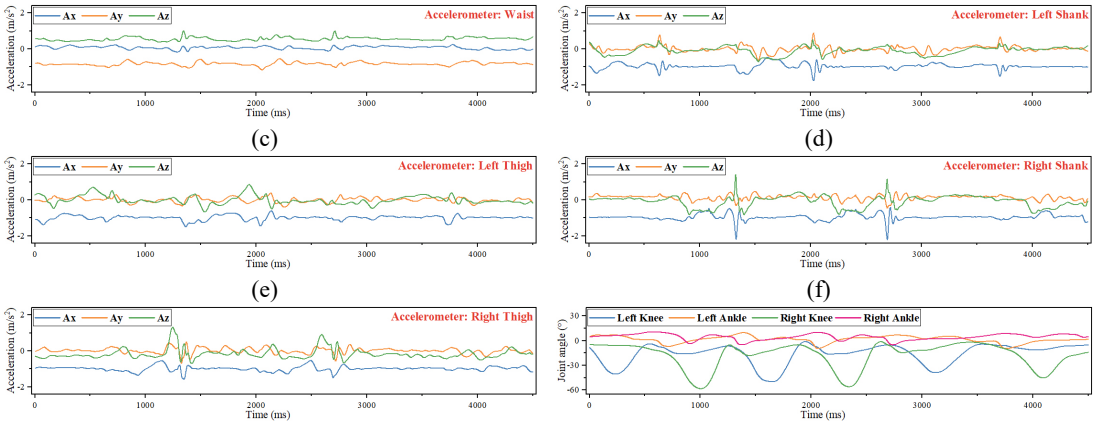


Figure 2. Time-domain waveforms of multimodal signals acquired by subject S_1 under the motion type of LGW in the first trail: (a) and (b) are sEMG signals of the left and right legs, respectively; (c)-(g) are the ACC signals of waist, left shank, left thigh, right shank and right thigh respectively; (h) shows GON signals of the knee and ankle joints of bilateral lower limbs.

(2) Dataset of Patients with Stroke

In this case, the sEMG and kinematic data of bilateral lower limbs of seven patients with stroke (denoted as P_1-P_7) were synchronously collected during rehabilitation training. The inclusion and exclusion criteria for patients can be found in our previous studies [34], details are presented in Table 1, where FMA-LE stands for Fugl-Meyer assessment of lower extremity. During each training session, they were required to perform three tasks, with an interval of 5-10 seconds between tasks to reduce muscle fatigue: Task 1 was to walk on a level ground for about one minute at a comfortable pace (Gait); Tasks 2 and 3 required subjects to repeat the SitTS and stand-to-sit processes approximately 10 times in an adjustable height chair without any assistance. All experimental procedures were approved by the ethical committee of Tongji Medical College of Huazhong University of Science and Technology (No. [2020] S296-1).

Table 1. Demographic and clinical characteristics of the subjects.

Subject	Gender	Age (years)	Height (cm)	Weight (kg)	Stroke Type	Paretic Side	FMA-LE Score (0-34)	Days from Stroke Onset
P_1	M	45	175	70	HS	R	14	71
P_2	F	54	160	60	IS	R	10	40
P_3	M	53	176	73	HS	R	14	50
P_4	M	49	173	66	IS	L	12	44
P_5	M	47	170	64	HS	R	12	70
P_6	M	45	168	62	IS	L	16	23
P_7	F	33	160	40	IS	R	12	90
Mean (SD)	/	46.6 (6.5)	168.9 (6.2)	62.1 (9.9)	/	/	12.9 (1.8)	55.5 (21.0)

* M/F, Male/Female; H/I, Hemorrhagic Stroke/Ischemic Stroke; L/R, Left/Right; SD, Standard Deviation.

The wireless Ultimu EMG system (Noraxon USA Inc., Scottsdale, AZ, USA) was used to collect the sEMG data of each subject's bilateral lower limbs at a sampling frequency of 2 kHz, including: BF, Gluteus Medius (GM), MG, RF, ST, SOL, TA and VM. sEMG electrodes were

placed parallel to the muscle fibers and the skin was cleaned with alcohol wipes to reduce impedance. Kinematic data were collected synchronously using the IMU system (Noraxon USA Inc., Scottsdale, AZ, USA) at a sampling frequency of 200 Hz, and seven IMU sensors were placed on each subject's pelvis, bilateral thighs, calves, and feet. It was worth mentioning that the angle data of the hip, knee and ankle joints of the subjects' bilateral lower limbs were obtained by software post-processing (MR3 myoMUSULETM; Noraxon USA Inc., Scottsdale, AZ, USA). To be consistent with Case I, the sEMG and kinematic data were down-sampled and up-sampled to 1 kHz, respectively.

2.2 Preprocessing

(1) Data Normalization

Given the re-sampled multi-channel signal $x_m(i, j)$ of modality m, where i = 1, 2, ⋯, C_m, and C_m denotes the number of channels for modality m; j = 1, 2, ⋯, L, and L denotes the signal length per channel; m = 1, 2, ⋯, M, and M denotes the number of modalities. The Min-Max normalization technique is used to normalize $x_m(i, j)$ to the range of [-1, 1], as follows:

$$y_m(i,j) = \frac{2 \cdot \left(x_m(i,j) - \min_{i,j} x_m(i,j) \right)}{\max_{i,j} x_m(i,j) - \min_{i,j} x_m(i,j)} - 1. \tag{1}$$

This type of normalization preserves the relative size between the data of different channels in the same modality, and helps to improve the convergence speed and stability of the model. Then, the normalized data of different modalities are concatenated in the channel dimension to obtain the data $Y = [y_1; y_2; \cdots; y_M]$ for the subsequent sample segmentation.

(2) Sample Segmentation

As shown in Fig. 3, there are two types of sliding windowing methods for segmenting sample set, namely overlapping windowing and non-overlapping windowing, and the former is adopted in this study to ensure full utilization of limited information.

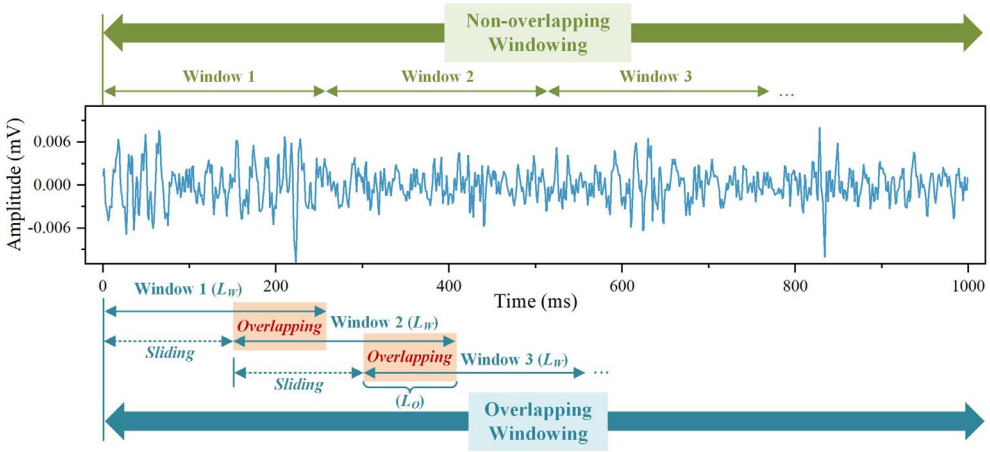


Figure 3. Schematic diagram of two different sliding windowing techniques.

Previous study has pointed out that there is an overall positive correlation between analysis window length (L_W) and classification accuracy [15]. Additionally, Naik et al. suggested that to ensure the comfort of human-robot interaction and the real-time nature of control, the overall

system delay should not exceed 300 ms [20]. Therefore, for the motion classification task, the L_W and overlapping window length (L_O) are initially set to 256 ms and 25%×L_W (*i.e.*, 64 ms) respectively referring to existing studies [11, 20, 30], and the impact of different L_O values on test accuracy will be discussed in Section IV. For the phase recognition task in Case II, due to the short duration of each phase in the SitTS process, a lower processing delay is required, so the L_W is set to 32/48/64 ms, with L_O set to 25%×L_W.

(3) Phase Segmentation of SitTS Motion

The SitTS motion is essential for achieving walking and other mobility activities, and is often impaired and not readily recoverable after stroke [34, 35]. Therefore, the analysis of complex SitTS motion in patients with early subacute stroke is conducive to active rehabilitation training and exploration of potential neurobiological mechanisms. In this study, the event detection and phase segmentation of the SitTS motion are referred to the previous literature [35]. As shown in Fig. 4, a complete SitTS motion contains five events: 1) Event$_0$, start of SitTS; 2) Event$_1$, start of seat-off; 3) Event$_2$, end of momentum transfer; 4) Event$_3$, start of stabilization; 5) Event$_4$, end of SitTS. Two adjacent events constitute a motion phase. Fig. 5 presents the event detection results for the first SitTS motion of subject P_4, and it should be noted that the outcomes are from the joint angle data of the affected limb, and the same is true for other subjects. Additionally, incomplete or abnormal STS motions are excluded by the visually identified 3D avatar animation generated by MR3 myoMUSULETM software [34].

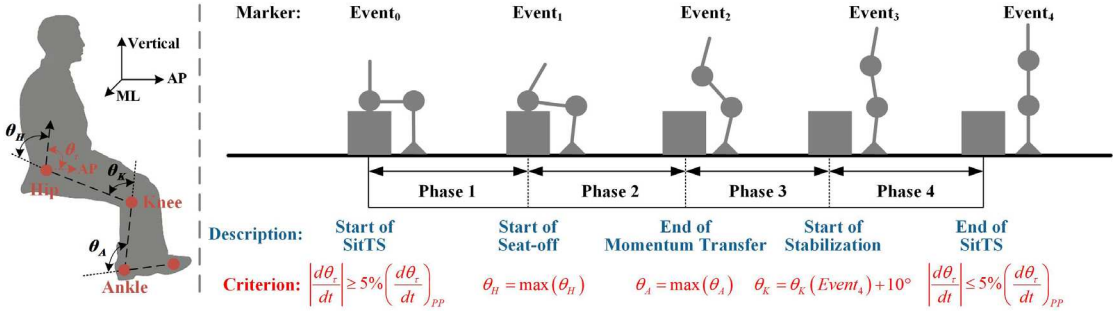


Figure 4. Event detection criteria for the SitTS motion of patients with stroke, where θ_T denotes the torso angle; θ_H, θ_A, and θ_K denote the hip angle, ankle angle, and knee angle of the affected limb, respectively, and they are complementary to the definitions in Ref. [35]; AP and ML denote the anterior-posterior and medial–lateral directions, respectively; PP denotes the Peak-to-Peak value.

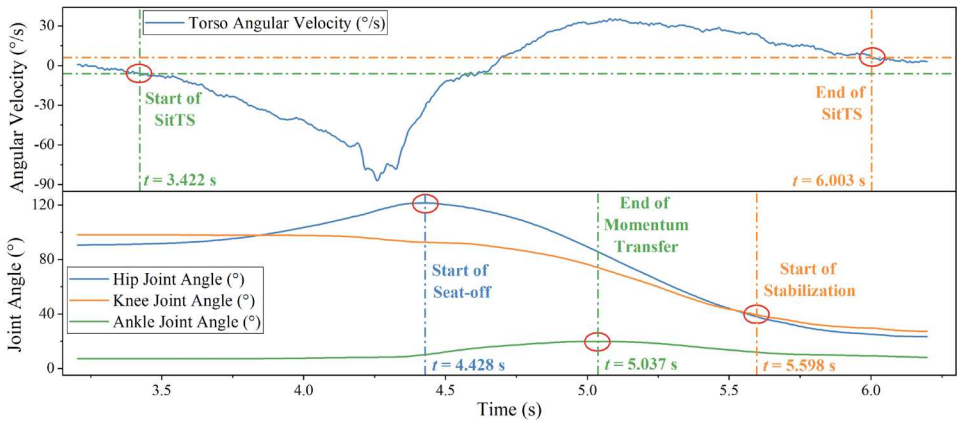


Figure 5. Event detection results for the first SitTS motion of subject P_4 (the affected limb).

(4) Dataset Partitioning

In this study, three scenarios are defined: 1) Intra-S, *i.e.*, subject-independent training and testing; 2) Default Inter-S, *i.e.*, directly using the data of target domain subjects to test the pre-trained model; 3) Inter-S with AdaBN or model fine-tuning, aiming to further enhance the domain generalization ability of pre-trained model. Taking ENABL3S dataset as an example, the partitioning of training set and test set is presented in Table 2. Additionally, in each trail, 10% of the training samples are randomly selected as the validation set to prevent overfitting.

Table 2. Partitioning of training set and test set under different scenarios on ENABL3S dataset.

Scenario Type	**Model Pre-training**	**Target Domain Model Tuning**	**Target Domain Model Test**
Intra-S	Without	Randomly select 80% of samples of S_i.	The remaining 20% of samples of S_i.
Inter-S (Default)	Randomly select samples of source domain subjects other than S_i.	Without	The remaining (1-β) of samples of S_i.
Inter-S (AdaBN / Model fine-tuning)		Randomly select a certain proportion (denoted as β) of samples of S_i.	

* S_i (i=1, 2, ⋯, 10) represents the target subject. The effect of the variations in β value on recognition performance and time cost will be analyzed in Section IV.

2.3 Deep Learning Models

(1) CNNs (CNN and Its Variants)

As one of the most representative DL algorithms, CNN and its variants have been widely applied in computer vision and other fields. As shown in Fig. 6(a), a basic CNN block designed in this study includes a convolutional (Conv) layer for feature extraction, a BN layer for accelerating model convergence, an activation function layer for enhancing model nonlinearity, a pooling layer for dimensionality reduction, and a dropout layer for preventing overfitting. As shown in Fig. 6(b), ResNet-V1 with a residual structure was proposed to solve the gradient vanishing and explosion problems in deep network training [36]. To reduce the parameter size, ResNet-V2 with a simplified network structure was further proposed [37]. As a representative of lightweight network, MobileNet-V1 replaced the standard Conv with depthwise separable Conv, greatly reducing the parameter size and operation cost [38], as shown in Fig. 6(c). Based on MobileNet-V1, a linear bottleneck block with inverted residual structure was proposed in MobileNet-V2 [39], as shown in Fig. 6(d). In MobileNet-V3 [40], the Squeeze-and-Excitation Network (SE-Net) [41] was inserted into the bottleneck block to make the module more focused on channel-sensitive features, and the Hard-Sigmoid and Hard-Swish activation functions were proposed to further lightweight the model, which are defined as follows:

$$\begin{aligned} Hard\text{-}Sigmoid[x] &= \begin{cases} 0 & , \ x < -2.5 \\ 0.2x + 0.5, & -2.5 \le x \le 2.5 \\ 1 & , \ x > -2.5 \end{cases} \\ Hard\text{-}Swish[x] &= x \cdot \frac{\mathrm{ReLU6}(x+3)}{6} = x \cdot \frac{\min(\max(0, x+3),\ 6)}{6} \end{aligned} . \quad (2)$$

ShuffleNet-V1 [42] and ShuffleNet-V2 [43] were another lightweight network architectures, as shown in Figs. 6(e) and 6(f), in which a channel shuffle operation was introduced to realize cross-channel information interaction. Additionally, ShuffleNet-V2 replaced the group Conv in ShuffleNet-V1 by a channel split operation.

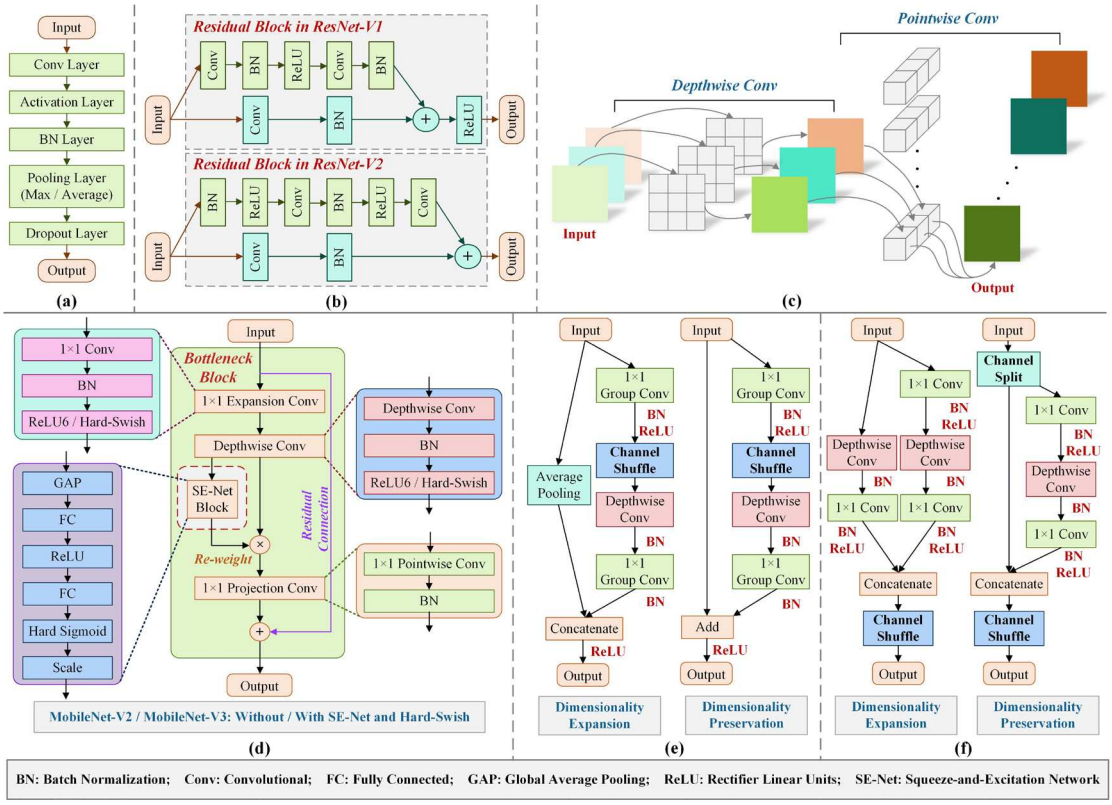


Figure 6. Different CNN-based blocks: (a) The designed basic CNN block; (b) Residual blocks in ResNet-V1 and ResNet-V2; (c) Depthwise separable Conv in MobileNet-V1; (d) Bottleneck blocks in MobileNet-V2 and MobileNet-V3; (e) Basic units in ShuffleNet-V1; (f) Basic units in ShuffleNet-V2.

(2) RNNs (RNN and Its Variants)

RNN is a commonly used network architecture for processing sequence data, with wide applications in tasks such as time series prediction. However, traditional RNN suffer from problems such as gradient vanishing and gradient explosion when dealing with long sequence data. In recent years, some variants of RNN have emerged, such as LSTM [25, 30, 31], BiLSTM [25], Gated Recurrent Units (GRU) [7] and Bidirectional GRU (BiGRU). LSTM introduces the gate mechanism and cell state to overcome the shortcoming of short-term dependence of RNN. However, it can only capture the influence of past states on future states. As an improvement, BiLSTM uses two-layer LSTM units to simultaneously perform forward and backward learning of sequence data. Compared to LSTM, GRU has a simpler structure and fewer parameters. Similarly, BiGRU is comprised of dual layers of GRUs. Detailed description of these RNN variants can be found in previous study [45].

(3) GNNs (GNN and Its Variants)

Different from CNNs, GNNs are proposed to process graph data in non-Euclidean domain,

whose core idea is to iteratively aggregate and propagate feature information between nodes through edge connections. Graph data can be expressed as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \{v_1, v_2, \cdots, v_n\} \in \mathbb{R}^{n \times d}$ is the set of nodes, n is the number of nodes, and d is the feature dimension of nodes; $\boldsymbol{E} = \{(v_i, v_j) | i, j \in [1, n]\}$ is the set of edges; $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the adjacency matrix representing the connection relationship between nodes, where $A_{i,j} = 1$ means $(v_i, v_j) \in \boldsymbol{E}$, otherwise $A_{i,j} = 0$.

Currently, GNNs have become one of the most prominent frameworks in the fields such as social networks, bioinformatics and fault diagnosis [46], which can be divided into spectral GNNs and spatial GNNs. The former achieve graph Conv through graph Fourier transform and inverse transform, while the latter perform Conv directly on the neighborhood of nodes. In this study, representatives of spectral GNNs and spatial GNNs, namely Chebyshev Network (ChebNet) [47] and Graph Attention Network (GAT) v2 [48], are chosen to construct the CNN-GNNs framework. The core of ChebNet is to approximate the spectral graph Conv by truncated expansion of the Chebyshev polynomial, thereby reducing computational complexity. The recurrence of the Chebyshev polynomial can be expressed as follows:

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),\ k \in N^+, \tag{3}$$

where $T_0 = 1$ and $T_1 = x$. Then, the kernel of spectral graph Conv can be expressed as follows:

$$g_\theta(\boldsymbol{\Lambda}) = \sum_{k=0}^{K-1} \theta_k T_k(\bar{\boldsymbol{\Lambda}}),\ \bar{\boldsymbol{\Lambda}} = 2\boldsymbol{\Lambda}/\lambda_{\max} - \mathbf{I_n}, \tag{4}$$

where $\boldsymbol{\Lambda}$ represents the diagonal matrix formed by the eigenvalues of Laplacian matrix $\mathbf{L}$; $\lambda_{\max}$ represents the largest eigenvalue in $\mathbf{L}$; $\mathbf{I_n}$ represents the identity matrix of dimension n; $\theta \in \mathbb{R}^K$ represents the coefficient vector of the Chebyshev polynomial. GATv2 is an improvement on the original GAT, introducing static attention and dynamic attention in the process of node aggregation. The updating process of node features can be expressed as follows:

$$h_i' = \sigma\left(\alpha_{i,i}\mathbf{W}h_i + \sum_{j \in N_i} \alpha_{i,j}\mathbf{W}h_j\right), \tag{5}$$

where h_i and h_i' represent the raw and updated features of node i, respectively; N_i represents the set of neighbor nodes of i; $\mathbf{W}$ is the weight matrix and σ is the activation function; $\alpha_{i,j}$ represents the attention weights between nodes i and j, generated by the multi-head attention [49].

2.4 Attention Mechanisms

(1) Coordinate Attention for CNNs

SE-Net was one of the most representative attention mechanisms in CNNs, but it neglected the modeling of location information [41]. Coordinate Attention Network (CA-Net) [50] embedded the location information into the channel attention, which enabled the network to acquire larger range of information while adding less computation.

As shown in Fig. 7(a), for an input feature map $x_c(i, j)$ of size $H \times W \times C$, two spatial extents of pooling kernels $(H, 1)$ or $(1, W)$ are respectively used to aggregate the location information of each channel along the horizontal and vertical coordinates, and then a pair of direction-aware

feature maps are yielded. These two transformations enable the module to capture long-range dependencies along one spatial direction and preserve position information along another direction. Then, these two transformations are concatenated in spatial dimension, and a shared 1×1 Conv is used to reduce the channel dimension to $max(8, C/r)$, where r is a reduction ratio. Next, the spatial information in vertical and horizontal directions is encoded using BN and non-linear. Afterwards, the encoded information is split, and two 1×1 Conv transforms are used to upgrade the channel number of two split feature maps to C, and then two sigmoid functions are used to generate normalized weights. Finally, the output $y_c(i, j)$ can be obtained as below:

$$y_c(i,j) = x_c(i,j) \cdot g_c^h(i) \cdot g_c^w(j), \tag{6}$$

where $g_c^h(i)$ and $g_c^w(j)$ represent the attention weights of two spatial directions, respectively.

(2) Multi-head Attention for RNNs and GATv2

Multi-head attention is a self-attention mechanism, as shown in Fig. 7(b). For the input data $\mathbf{X} \in \mathbb{R}^{n \times d}$, a series of linear transformations are first performed to obtain the projections of query (**Q**), key (**K**) and value (**V**): $\mathbf{Q}_i = \mathbf{X}\mathbf{W}_i^{\boldsymbol{Q}}$, $\mathbf{K}_i = \mathbf{X}\mathbf{W}_i^{\boldsymbol{K}}$, $\mathbf{V}_i = \mathbf{X}\mathbf{W}_i^{\boldsymbol{V}}$, where $i \in \{1, 2, \cdots, N_h\}$ and N_h is the number of heads; $\mathbf{W}_i^{\boldsymbol{Q}}, \mathbf{W}_i^{\boldsymbol{K}} \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}_i^{\boldsymbol{V}} \in \mathbb{R}^{d \times d_v}$ are learnable weight matrices. Then, these projections are entered into the scaled dot-product attention [48] to obtain the attention score of the i-th head:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i). \tag{7}$$

Finally, concatenate the output of each head and perform another linear transformation to obtain the final multi-head attention, expressed as follows:

$$\text{Multi-head}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \cdots, \text{head}_{N_h})\mathbf{W}^{\boldsymbol{O}}, \tag{8}$$

where $\mathbf{W}^{\boldsymbol{O}} \in \mathbb{R}^{N_h d_v \times d}$ is a learnable weight matrix. In this study, N_h is set to 4 considering the computational complexity, and $d_k = d_v = d/4$.

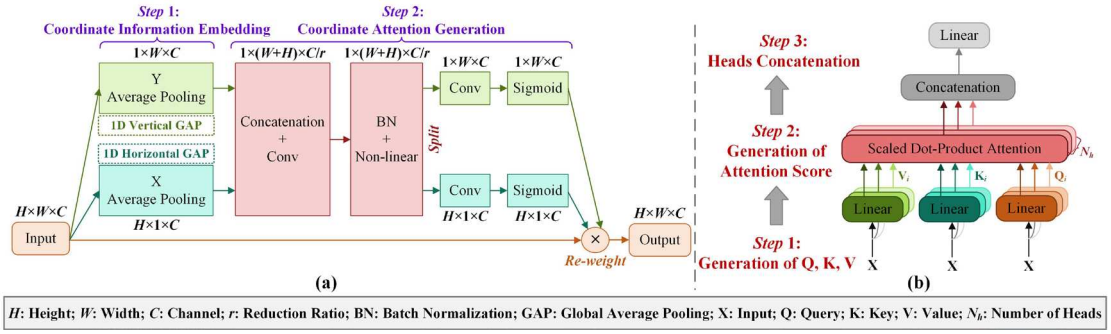


Figure 7. Structure diagrams of different attention mechanisms: (a) CA-Net; (b) Multi-head attention.

2.5 Evaluation Framework

(1) Classification Performance

For class-imbalanced multi-classification tasks, it is difficult to objectively reflect the test

results by only using accuracy. Thus, evaluation of classification performance involves metrics such as accuracy, F1-Score (macro-averaged), and normalized confusion matrix.

(2) Model Complexity

The complexity of DL models involves space complexity and time complexity. In this study, the former is quantified by the number of parameters, while the latter is estimated by a commonly used metric, the Multiply-Adds (M-Adds) [38-40, 50].

(3) Model Adaptability

Model adaptability reflects the impact of variations in the number of modalities or sensor channels on network structure and hyperparameters. Since it cannot be quantified, this study adopts subjective evaluation criteria to divide it into three grades: weak, medium and strong.

3. Proposed Frameworks

3.1 Graph Data Construction in CNN-GNNs Framework

In the CNN-GNNs framework, multimodal signals need to be converted into graph data to serve as input to GNNs. Specifically, each channel of multimodal signal is regarded as a node to obtain the set of nodes (***V***); The shallow features extracted by CNN are regarded as node features to obtain the feature matrix (***X***); The connection relationships between channels (or nodes) are regarded as the set of edges (***E***), and then the adjacency matrix (***A***) can be obtained. Describing the relationships between signals of different channels in intra-modal and inter-modal scenarios as reasonably as possible is the key to constructing ***A***. Next, the ENABL3S dataset is used as an example to illustrate the proposed graph data construction methods, and it should be noted that it is stipulated that each node may have a self-connected edge.

(1) Generation of KNNG without Master Nodes

Currently, most studies construct graph topology by measuring the correlation or distance between node features [51]. The KNNG uses Euclidean distance to judge the adjacency relationship between nodes, and its distance metric can be expressed as:

$$D_{i,j} = \sqrt{\sum_{l=1}^{d} \left| v_i^{(l)} - v_j^{(l)} \right|^2}, \tag{9}$$

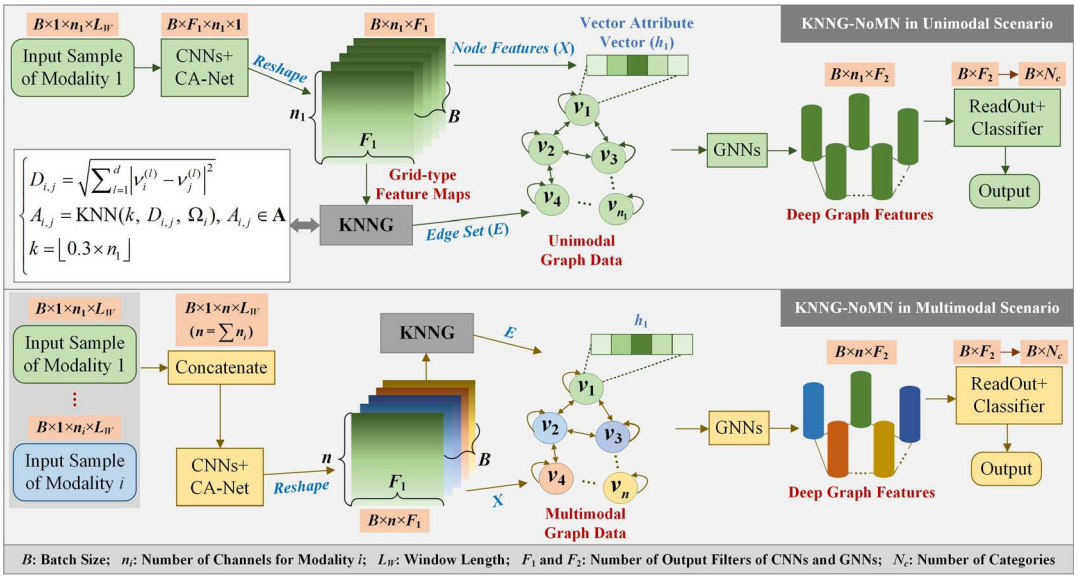
where $D_{i,j}$ represents the distance between nodes v_i and v_j; d represents the feature dimension of v_i and v_j; $v_i^{(l)}$ and $v_j^{(l)}$ represent the l-th eigenvalue of v_i and v_j, respectively. The edge construction of KNNG can be expressed as:

$$A_{i,j} = \mathrm{KNN}(k,\ D_{i,j},\ \Omega_i),\ A_{i,j} \in \mathbf{A}, \tag{10}$$

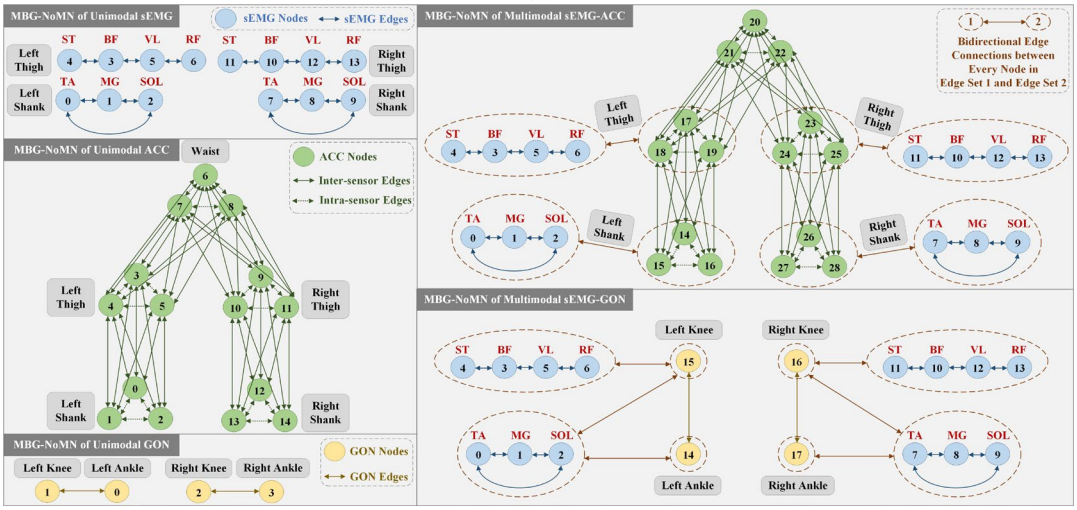
where $\Omega_i = \{D_{i,1}, D_{i,2}, \cdots, D_{i,n}\}$ represents the set of distance between v_i and all nodes; k is a hyperparameter representing the number of nearest neighbors, and k is initially set to $\lfloor 0.3 \times n \rfloor$ in this study, where n is the number of channels (nodes). If $D_{i,j}$ is the minimum of k in Ω_i, $\mathrm{KNN}(\cdot) = 1$, otherwise $\mathrm{KNN}(\cdot) = 0$. Fig. 8(a) shows the construction process of KNNG without master nodes (denoted as KNNG-NoMN) in unimodal and multimodal scenarios.

(2) Generation of MBG without Master Nodes

Another common method of graph topology generation is based on human physiological structure and spatial placement of sensors. For example, Wu et al. [26] and Massa et al. [52] constructed $\boldsymbol{A}$ based on the physiological connection between the four joints of bilateral lower limbs and the spatial relationship of high-definition sEMG sensors, respectively. Inspired by these studies, the construction process of MBG without master nodes (denoted as MBG-NoMN) in unimodal and multimodal scenarios is shown in Fig. 8(b): 1) When only unimodal sEMG data or kinematic (ACC or GON) data is used, the edge connection of MBG-NoMN is similar to the above studies, which is determined by the physiological structure of lower limb muscles and the spatial relationship between sensors; 2) When multimodal sEMG data and kinematic data are used simultaneously, the cross-modal edge connections of MBG-NoMN are determined by the biomechanical properties of lower limb joint motion; 3) It is specified that each node has a self-connected edge (not drawn in Fig. 8(b)).



(a)



(b)

Figure 8. Different graph topology construction processes in unimodal and multimodal scenarios: (a) KNNG-NoMN; (b) MBG-NoMN.

(3) Master Nodes-based Cross-modal Information Interaction

Usually, the signal characteristics of the same modality are similar, while for different modalities they may vary greatly. Consequently, for graph topology determined by similarity or distance, edge connections may only exist between nodes of the same modality, and there may be no inter-modal information flow. In terms of this issue, this study further proposes graph data construction methods based on master nodes in multimodal scenarios, denoted as KNNG-MN and MBG-MN, respectively. Taking any two modal information fusion as an example, the main steps are described below, and it can be easily extended to multimodal signals.

Step 1: For normalized data y_m of modality m (m = 1, 2), where the number of channels is n_1 and n_2, respectively, using CNN to exact node features to obtain feature matrices $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, and then the sets of nodes $\boldsymbol{V_1} = \{v_1, \cdots, v_{n_1}\}$ and $\boldsymbol{V_2} = \{v_{n_1+1}, \cdots, v_{n_1+n_2}\}$ can be obtained.

Step 2: Generate the sets of intra-modal edge connections $\boldsymbol{E_1} = \{(v_i, v_j)|i, j \in [1, n_1]\}$ and $\boldsymbol{E_2} = \{(v_i, v_j)|i, j \in [n_1+1, n_1+n_2]\}$ according to KNNG or MBG.

Step 3: Add the set of master nodes $\boldsymbol{V_3} = \{v_{n_1+n_2+1}, v_{n_1+n_2+2}\}$, and initialize the node feature matrix $\boldsymbol{X_3}$ to 0, and then the total set of nodes $\boldsymbol{V} = \boldsymbol{V_1} \cup \boldsymbol{V_2} \cup \boldsymbol{V_3}$ can be obtained.

Step 4: Add the edge connections between master nodes and nodes of specific modalities, denotes as $\boldsymbol{E_3} = \{(v_i, v_j)|(i \text{ or } j = [1, n_1], j \text{ or } i = n_1+n_2+1), (i, j = n_1+n_2+1)\}$ and $\boldsymbol{E_4} = \{(v_i, v_j)|(i \text{ or } j = [n_1+1, n_1+n_2], j \text{ or } i = n_1+n_2+2), (i, j = n_1+n_2+2)\}$, and inter-modal edge connections $\boldsymbol{E_5} = \{(v_i, v_j)|i, j = [n_1+n_2+1, n_1+n_2+2]\}$, and then the total set of edges $\boldsymbol{E} = \boldsymbol{E_1} \cup \boldsymbol{E_2} \cup \boldsymbol{E_3} \cup \boldsymbol{E_4} \cup \boldsymbol{E_5}$ can be obtained, as well as the graph data $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$.

Step 5: In the update process of GNNs, the master nodes iteratively aggregate and propagate feature information through intra-modal and inter-modal edge connections, thereby realizing the dynamic interaction and fusion of multimodal information.

3.2 Four Multimodal Information Fusion and LLMR Frameworks

Based on the idea of multimodal and multi-sensor information fusion, four different LLMR frameworks are presented, as shown in Fig. 9. Taking the fusion of sEMG and ACC data on ENABL3S dataset as an example, their main architectures are outlined as follows:

1) CNNs-V1 framework: Firstly, the samples of different modalities are concatenated in the dimension of sensor channel. Then, four 1D CNN blocks are used for time step-wise feature extraction, and two 1D CNN blocks are used for deep sensor channel-wise information fusion. Finally, the LLMR results are output by the classifier. In addition, the CNN blocks can be easily replaced with the other seven variant blocks shown in Fig. 6.

2) CNNs-V2 framework: Three 2D CNN blocks are used for simultaneous feature extraction in both time step-wise and sensor channel-wise, mainly reflected in the change of parameters such as kernel size and stride of the Conv and pooling layers.

3) CNN-RNNs framework: Three 1D CNN blocks and CA-Net are used for spatial feature extraction, and an RNN layer and a multi-head attention layer are used for temporal relationship modeling, and then the feature maps that aggregate the spatial-temporal information are used

for downstream classification tasks.

4) CNN-GNNs framework: Similar to CNN-RNNs, four 1D CNN blocks and CA-Net are first used for shallow feature extraction, then the grid-type feature maps are embedded into the graph data combined with KNNG or MBG methods, and finally GNNs are used to extract deep graph features that aggregate global information for downstream classification tasks.

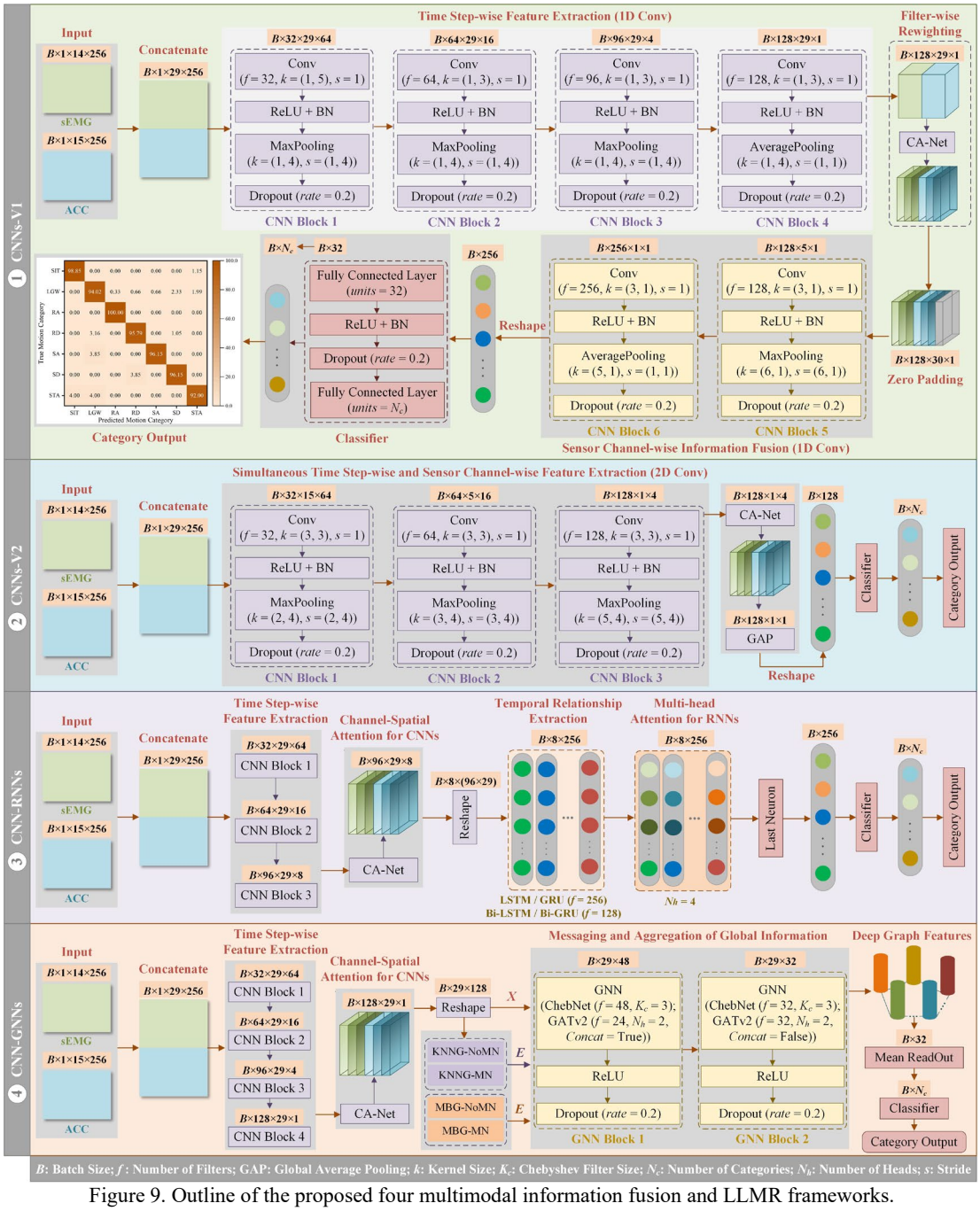


Figure 9. Outline of the proposed four multimodal information fusion and LLMR frameworks.

3.3 Transfer Learning Techniques for Inter-S Prediction

In this study, AdaBN and model fine-tuning are employed to further enhance the domain generalization ability of the proposed LLMR frameworks in Inter-S prediction. Taking CNN-RNNs model as an example, Table 3 presents the main flow of AdaBN [29].

Table 3. AdaBN-based model parameter update procedure.

Algorithm 1: AdaBN-based Model Parameter Update	
Input:	Target domain data d; i-th neuron of a BN layer of the pre-trained CNN-RNNs model $x_t^i(d) \in \mathbf{x}_t^i$, where $\mathbf{x}_t^i = \left\{x_t^i(1), \cdots, x_t^i(n)\right\}$, and the scaling factor γ_s^i and the shift factor β_s^i of the i-th neuron.
Output:	Updated target domain CNN-RNNs model.
For	Each neuron i and input data d in target domain, **do**:
	Calculate the mean and variance of all samples in target domain: $\mu_t^i \leftarrow \mathbf{E}[\mathbf{x}_t^i]$, $\sigma_t^i \leftarrow \sqrt{\mathbf{Var}[\mathbf{x}_t^i]}$;
	Calculate the output of the BN layer: $\hat{\mathbf{x}}_t^i(d) = (x_t^i(d) - \mu_t^i)/\sigma_t^i$, $y_t^i(d) = \gamma_s^i \hat{\mathbf{x}}_t^i(d) + \beta_s^i$.
End for	

For the model fine-tuning, the weights of the pre-trained CNN-RNNs are assigned to a new CNN-RNNs, and then the new model is retrained using the target domain data until it converges. Specifically, a smaller initial learning rate (10^{-4}) is used to fine-tune the feature extraction layers, aiming to preserve the learned general feature representations as much as possible. Meanwhile, a larger initial learning rate (10^{-3}) is used to train the classifier, aiming to make it adapt to the target domain data more quickly.

4. Results and Discussion

The algorithm design was implemented using Python 3.9.18 and PyTorch 1.13.0 DL libraries. Experiment platform was a high-performance server configured with GeForce RTX 3080Ti GPU, Intel® Xeon® CPU E5-2680 v4@ 2.40 GHz and 32G RAM.

4.1 Case I: Experimental Results on ENABL3S Dataset

4.1.1 In Intra-S Scenario

In this subsection, the partitioning of training set, validation set and test set is described in Subsection 2.2.4. The Adam algorithm is employed as the optimizer, with an initial learning rate of 10^{-2}, and the batch size and maximum training epochs is set to 32 and 100, respectively. In addition, the model training process is optimized by employing learning rate decay (ReduceLROnPlateau) and early stopping techniques. For each subject, five replicate experiments are performed, and each experiment will yield a different random number seed to ensure a different sample set partitioning result, so as to minimize the effect of accidental factors. In other words, for each model, a total of 50 test results will be yielded on 10 subjects. Table 4 presents the means and SDs of test results for different models with the fusion of sEMG and ACC data, as well as the corresponding model complexity and adaptability.

As can be observed from Table 4: 1) In CNNs-V1 framework, ResNet-V1 has the highest recognition accuracy (94.844±0.639%) and F1-Score (93.717±0.740%), as well as the highest complexity. MobileNet-V1 has the lowest complexity and classification performance, while ShuffleNet-V1 has a slight increase in complexity but also an improvement in classification performance. 2) Compared with CNNs-V1, the CNN-RNNs framework has better classification performance, especially the CNN-BiLSTM, reaching a mean accuracy of 95.198%, but RNNs

also bring more model parameters; 3) Compared with CNN-RNNs, the complexity of CNN-GNNs framework is greatly reduced. For the two GNNs, the classification performance of MBG-NoMN is close to that of MBG-MN, indicating the rationality of MBG-NoMN to set the inter-modal edge connections according to the biomechanical properties of muscles. Compared with KNNG-NoMN, the classification performance of KNNG-MN is greatly improved, indicating that the proposed master nodes-based graph data construction method can effectively promote the interactive fusion of sEMG and ACC data. Combined with MBG-MN, CNN-ChebNet achieves the highest mean accuracy of 94.035%.

Table 4. Evaluations of classification performance using sEMG-ACC data on ENABL3S dataset, as well as complexity and adaptability of different models.

Framework / (Adaptability)	**Block**	**Graph Topology**	**Classification Performance**		**Model Complexity**	
			Accuracy (%) (Mean ± 1 SD)	**F1-Score (%) (Mean ± 1 SD)**	**Number of Parameters**	**M-Adds (10^6)**
CNNs-V1 / (Weak)	Basic-CNN-V1	/	94.433±0.584	92.980±0.770	223,053	27.882
	ResNet-V1	/	**94.844±0.639**	**93.717±0.740**	614,253	79.738
	ResNet-V2	/	94.014±0.720	92.339±1.154	612,717	79.737
	MobileNet-V1	/	93.485±0.840	91.375±1.095	**85,581**	**10.836**
	MobileNet-V2	/	94.359±0.528	92.817±0.896	395,405	47.210
	MobileNet-V3	/	94.493±0.804	93.043±1.228	415,901	46.370
	ShuffleNet-V1	/	94.131±0.889	92.628±1.070	88,565	11.029
	ShuffleNet-V2	/	93.861±1.150	92.253±1.432	195,533	22.427
CNN-RNNs / (Strong)	LSTM	/	95.008±0.976	93.393±1.360	3,348,749	46.731
	BiLSTM	/	**95.198±0.839**	**93.740±1.075**	3,217,677	45.683
	GRU	/	94.717±0.771	92.876±0.935	2,569,997	40.501
	BiGRU	/	94.895±0.773	93.299±1.061	**2,471,693**	**39.715**
CNN-GNNs / (Medium)	ChebNet	KNNG-NoMN	91.209±2.102	86.627±2.845	106,525	26.395
		KNNG-MN	93.625±1.609	91.303±1.871		
		MBG-NoMN	93.988±1.027	91.485±1.890		
		MBG-MN	**94.035±1.411**	**91.611±1.886**		
	GATv2	KNNG-NoMN	89.454±2.299	83.419±3.501	102,253	26.498
		KNNG-MN	93.471±0.937	90.541±1.868		
		MBG-NoMN	93.497±0.916	90.817±1.477		
		MBG-MN	**93.615±1.134**	**90.987±1.283**		

In addition, Fig. 10(a) presents the test accuracy distribution of using ResNet-V1, CNN-BiLSTM, and CNN-ChebNet (MBG-MN) in five replicate experiments for each subject via half-violin plots, while Fig. 10(b) specifically presents the confusion matrices of test results using these three models on subject S_1. For each half-violin plot, a bee colony plot reflecting the distribution location and number of data points distributed is shown on the left, a kernel density plot reflecting the probability density distribution of data points is shown on the right,

and the mean and SD of data points are shown on the middle right.

Experimental results indicate that the proposed three frameworks all perform well in the LLMR task of healthy subjects. When both classification performance and model complexity are considered, the Basic-CNN-V1 and ShuffleNet-V1 strike a good balance, while CNN-BiLSTM may be a better choice when the latter is ignored. In terms of model adaptability, CNNs-V1 and CNNs-V2 need to design the size of Conv and pooling kernels according to the number of sensor channels and sample length, especially for CNNs-V2, whose adaptability is the weakest; CNN-RNNs and CNN-GNNs are not affected by the change in the number of channels, but the latter needs to consider the construction of graph data. In practical applications, specific requirements such as recognition accuracy and hardware performance need to be considered comprehensively to select the most suitable LLMR model.

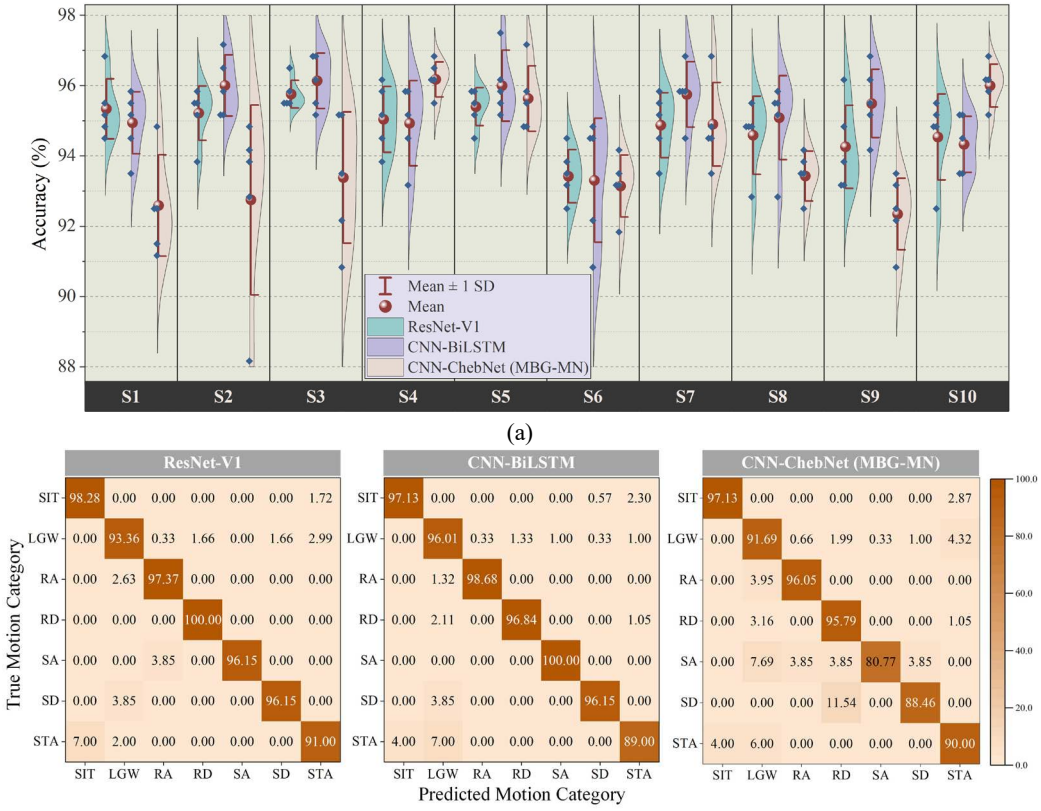


Figure 10. Experimental results on ENABL3S dataset: (a) Test accuracy distribution in five replicate experiments for each subject using different models; (b) Confusion matrices of test results using different models on subject S_1.

4.1.2 In Inter-S Scenario

In this subsection, taking CNN-BiLSTM as the model and S_1 as the target subject, the experimental setups are as follows: 1) Randomly select five subjects other than S_1 as source domain subjects, and use the source domain data to pre-train CNN-BiLSTM according to the settings in Subsection 4.1.1; 2) For AdaBN and model fine-tuning, a certain proportion (the β shown in Table 2) samples of S_1 are used to further adjust the pre-trained CNN-BiLSTM according to the settings in Subsection 3.3, while no operation is required for the default Inter-S scenario; 3) The remaining (1-β) samples of S_1 are used to test the final CNN-BiLSTM. Fig.

11(a) presents the means and SDs of the classification performance metrics in five replicate experiments on S_1 at different β values, and Fig. 11(b) shows the corresponding time costs.

As can be observed from Fig. 11(a): 1) The mean recognition accuracy of the default Inter-S scenario is more than 80% and is almost unaffected by the change in β values; 2) The mean accuracy with AdaBN is slightly increased (0.556-1.002%), while it is significantly increased with model fine-tuning. Specifically, when $\beta \geq 0.1$, the mean accuracy has already exceeded 90%, and when $\beta \geq 0.6$, it surpasses 95%, outperforming the performance of CNN-BiLSTM in the Intra-S scenario shown in Fig. 10(a). As can be observed from Fig. 11(b): 1) The time cost of sampling is linearly related to the β value, and the time cost of AdaBN can be essentially ignored; 2) For model fine-tuning, a total time cost of at least 93.60 seconds (including sampling time and fine-tuning time) is needed to achieve a mean recognition accuracy of over 90%.

Experimental results indicate that despite differences in data distribution between target domain and source domain subjects, the pre-trained CNN-BiLSTM itself possesses a certain domain adaptation ability and achieves satisfactory recognition accuracy without the aid of additional techniques. Furthermore, after model fine-tuning, a modest time cost investment can significantly enhance the generalization performance of the pre-trained CNN-BiLSTM.

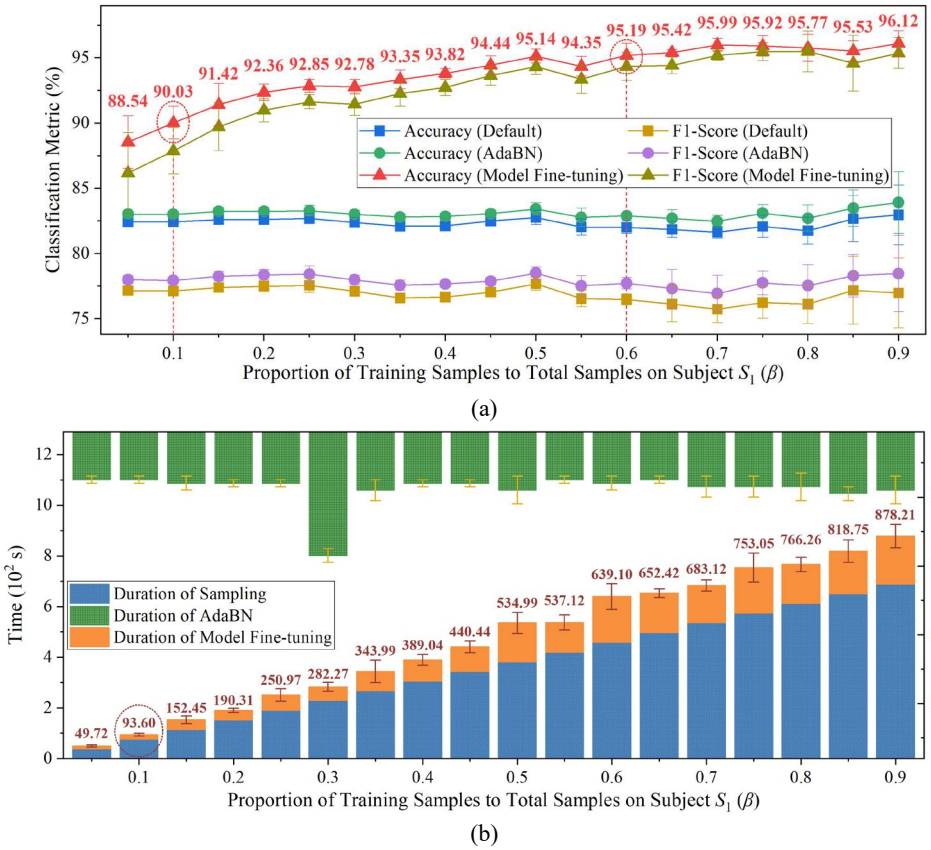


Figure 11. Experimental results using different techniques in Inter-S scenario (on target subject S_1): (a) shows the changes of classification performance metrics at different β values, and (b) shows the corresponding time costs.

4.1.3 Ablation Study

Firstly, taking Basic-CNN-V1 and CNN-BiLSTM as examples, the effect of sensor modalities on recognition performance is discussed in Intra-S scenario. Different modal

combinations are used to generate different data subsets, including unimodal data (sEMG, ACC and GON) and multimodal data (sEMG-ACC and sEMG-GON). For Basic-CNN-V1, the parameters of the pooling layers in CNN Blocks 5 and 6 (Fig. 9), including kernel size and stride, need to be adjusted according to the number of input channels, while no changes are required for CNN-BiLSTM. All other experimental setups refer to Subsection 4.1.1. Figs. 12(a) and 12(b) show the distribution of different classification performance metrics of both models in five replicate experiments across all subjects, with the top of each plot showing the mean corresponding to the half-violin plot. It can be observed that: 1) Among the three types of unimodal data, sEMG performs the worst while GON performs the best; 2) The recognition performance of CNN-BiLSTM is superior to that of Basic-CNN-V1; 3) The recognition performance using multimodal data is superior to that using any unimodal data, and for CNN-BiLSTM, the mean accuracy using sEMG-ACC data is improved by 3.195% and 0.805%, respectively, compared with using sEMG data and ACC data alone, while for sEMG-GON data, it is improved by 3.336% and 0.881%, respectively.

Experimental results show that it is feasible to use only kinematic data to achieve LLMR of healthy subjects, exhibiting superior recognition performance compared to using sEMG, which may be caused by some motion-related feedforward signals in IMUs and GONs. A similar phenomenon has been reported in Ref. [53], where the mean accuracy of using unimodal sEMG data alone is less than 85%. Therefore, from the perspective of reducing the cost of data acquisition, the recognition performance can also be basically satisfied using only motion sensors. However, due to the anticipatory nature of sEMG measurements, it may be more suitable for predicting continuous motion intentions in advance, such as joint angles [30] or joint torques [31]. Furthermore, combining kinematic data with sEMG data is beneficial to further improve recognition accuracy and enhance the robustness of the recognition system.

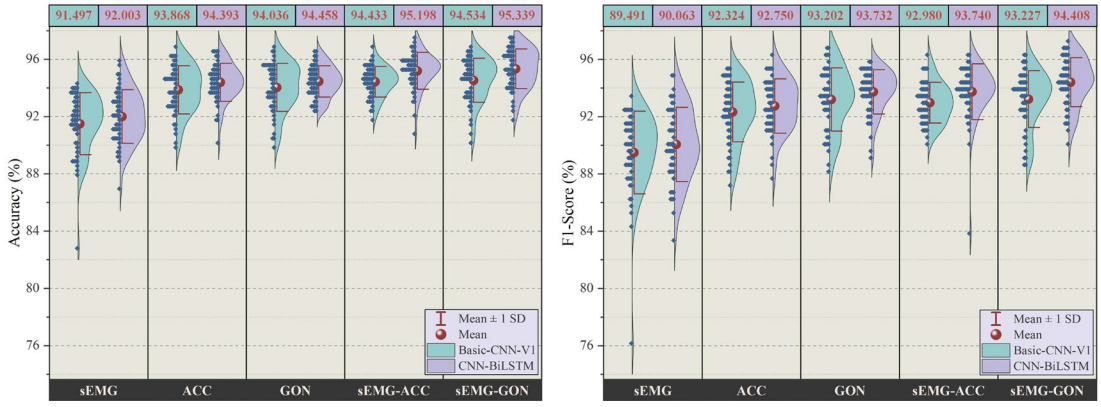
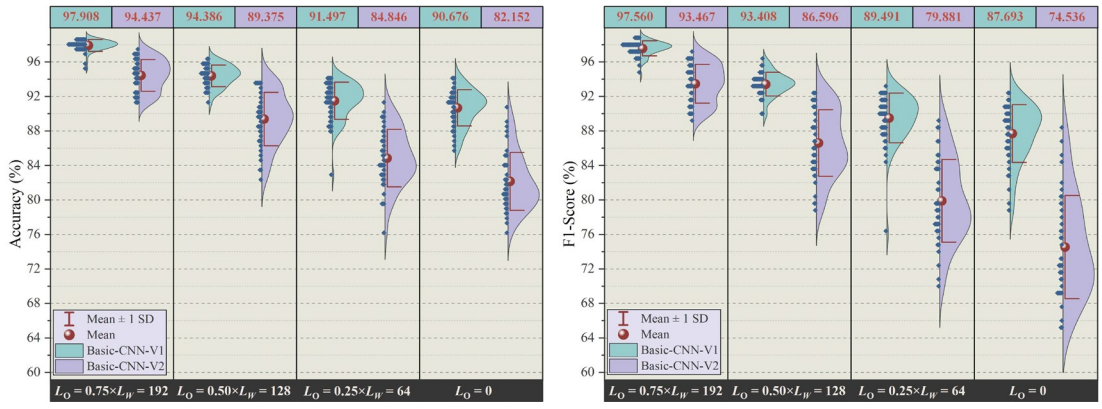


Figure 12. Distribution of different classification performance metrics in five replicate experiments across all subjects using different modal data (on ENABL3S dataset): (a) Accuracy; (b) F1-Score.

Then, taking the unimodal sEMG data as an example, the effect of different L_O values on recognition performance is analyzed in Intra-S scenario. The basic CNN blocks are used to construct CNNs-V1 and CNNs-V2 frameworks, to obtain the Basic-CNN-V1 and Basic-CNN-

V2 models, respectively. The L_W is maintained at 256 ms, and different data subsets are generated by setting different L_O values (0/64/128/192 ms). All other experimental setups refer to Subsection 4.1.1, and Figs. 13(a) and 13(b) present the distribution of different classification performance metrics of both models in five replicate experiments across all subjects. It can be observed that the recognition performance of Basic-CNN-V1 is far superior to Basic-CNN-V2, and as L_O decreases, the performance gap between the two becomes larger. In the case of non-overlapping windowing ($L_O = 0$), the mean accuracy of Basic-CNN-V1 still exceeds 90%, while the latter is only 82.152%. Specifically, as L_O changes from 192 ms to 0 ms, the mean accuracy and mean F1-Score of Basic-CNN-V1 decreased by 7.232% and 9.867%, respectively, while the latter decreased by 12.285% and 18.931%, respectively.

There may be two reasons for this phenomenon: 1) The number of training samples increases with the increase of L_O, which means that the model can be trained more sufficient; 2) Sample similarity will increase with the increase of L_O, *i.e.*, there may be more overlap between training and test samples. In this case, higher test accuracy is actually meaningless. Therefore, setting L_O to 64 ms can minimize sample similarity while expanding sample size, thus obtaining more reasonable test results. In addition, in the CNNs-V2 framework, feature extraction in time step-wise and sensor channel-wise is carried out simultaneously via 2D Conv. In contrast, in the CNNs-V1 framework, the two operations are separated, and deep sensor channel-wise information fusion is carried out in the second stage, which seems to be a more reasonable way of model construction. Experimental results may provide a new perspective for the construction of DL models based on multi-channel data input in the LLMR field.



(a) (b)

Figure 13. Distribution of different classification performance metrics in five replicate experiments across all subjects at different L_O values (on ENABL3S dataset): (a) Accuracy; (b) F1-Score.

4.1.4 Comparison with State-of-the-Art Methods

In this subsection, the proposed frameworks are compared with some state-of-the-art methods in the LLMR field. All comparative experiments uniformly use sEMG-ACC data and are conducted in Intra-S scenario according to the settings in Subsection 4.1.1. Specifically, this study compares several commonly employed shallow machine learning classifiers, namely KNN [10, 12, 17], LDA [12, 20, 23, 54], SVM [10, 12, 17, 54], and Random Forest (RF) [21].

For each classifier: 1) Six time-domain features are extracted per sEMG channel, including Root Mean Square (RMS), mean absolute value, waveform length, average amplitude change, zero-crossing rate, and Willison amplitude. Six time-domain features are extracted per ACC channel, including RMS, mean, maximum, minimum, peak-to-peak value, and SD; 2) In total, a feature subset of 174 dimensions is obtained and subjected to Min-Max normalization for its use as input to the classifier; 3) The classifier's hyperparameters are optimized by grid search algorithm. Experimental results are shown in Table 5, where SVM has the highest test accuracy (93.997±0.736%), and LDA performs the worst. These machine learning classifiers have lower complexity and faster computational efficiency than DL models. However, they rely on domain expertise for manual feature extraction and thus fail to achieve end-to-end LLMR.

In addition, this study also compares several DL models applied on ENABLE3S dataset, namely CNN2DLSTM [55], CNN2DGRU [55], MSC-CNN [56] and MCD [53]. Table 5 presents the number of parameters, M-Adds and test results of these models. In terms of test accuracy, CNN2DGRU performs the best but still slightly lower than the three proposed frameworks. In terms of model complexity, since MSC-CNN and MCD consider that pooling layers may lead to information loss, they only use Conv operations for feature dimension reduction, resulting in higher M-Adds values. In particular, MCD is an unsupervised DL model designed to achieve LLMR for target subjects whose signals are unlabeled.

Table 5. Comparison with state-of-the-art methods using sEMG-ACC data on ENABL3S dataset.

Type	Method	Number of Parameters	M-Adds (10^6)	Accuracy (%) (Mean ± 1 SD)
Shallow Machine Learning	KNN [10, 12, 17]	/	/	93.201±0.713
	LDA [12, 20, 23, 54]	/	/	89.400±1.254
	SVM [10, 12, 17, 54]	/	/	**93.997±0.736**
	RF [21]	/	/	93.484±0.957
Deep Learning	CNN2DLSTM [55]	49,877	21.392	93.470±1.439
	CNN2DGRU [55]	40,337	20.782	**94.024±1.499**
	MSC-CNN [56]	225,365	126.971	92.435±1.679
	MCD [53]	379,739	71.177	93.600±2.360
Proposed	ResNet-V1 (CNNs-V1)	614,253	79.738	94.844±0.639
	CNN-BiLSTM	3,217,677	45.683	**95.198±0.839**
	CNN-ChebNet (MBG-MN)	106,525	26.395	94.035±1.411

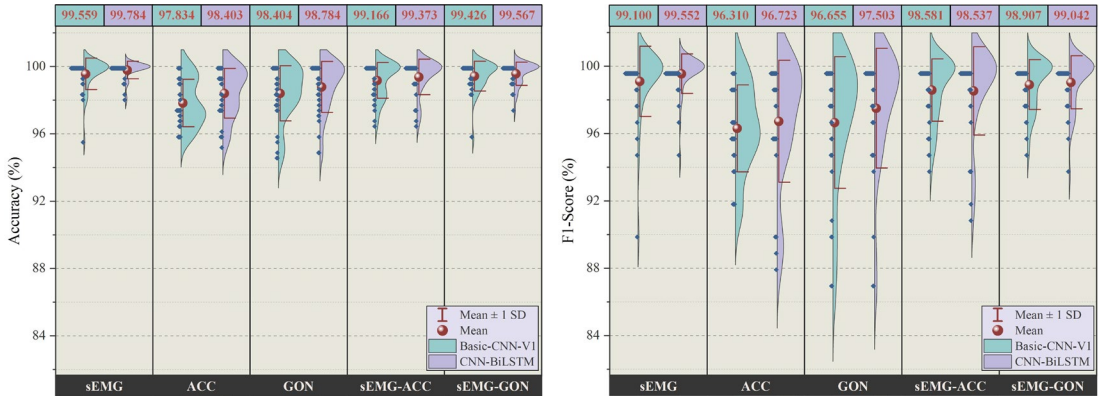
4.2 Case II: Experimental Results on Dataset of Patients with Stroke

4.2.1 Regarding the LLMR Task

In this subsection, Basic-CNN-V1 and CNN-BiLSTM are adopted to further investigate the applicability of the proposed LLMR frameworks on patients with stroke. The L_W and L_O are set to 256 ms and 64 ms, respectively, and the other experimental setups refer to Case I.

Similarly, the effect of sensor modalities on recognition performance is analyzed in Intra-S scenario, and the distribution of different metrics of both models in five replicate experiments across all subjects are shown in Figs. 14(a) and 14(b). It can be observed that: 1) CNN-BiLSTM performs slightly better than Basic-CNN-V1 regardless of the variation of sensor modalities; 2) Slightly different from the experimental results on healthy subjects in Case I (Fig. 12), in this case, the recognition performance of these two models using unimodal sEMG data is superior to that using unimodal kinematic data (*e.g.*, for CNN-BiLSTM, the mean accuracy improved by 1.725% and 1.155% compared to using ACC data and GON data), as well as surpassing the performance using multimodal data.

The reason for this phenomenon may be that only the affected limb data of each patient was used for experimental validation. Unlike healthy subjects, patients with stroke have heterogeneity regarding lower limb motor dysfunction, which leads to instabilities and irregularities in kinematic data. In such cases, fusing kinematic data may degrade the performance of sEMG. Experimental results demonstrate the potential of bioelectric sensors such as sEMG for LLMR applications in patients with lower limb motor dysfunction caused by neurological diseases such as stroke. However, this finding still requires data from additional patients for further verification. Additionally, for relatively simple classification tasks, further research can be conducted on the selection of sEMG channels to enhance the comfort of subjects and reduce system costs.



(a) (b)

Figure 14. Distribution of different classification performance metrics in five replicate experiments across all subjects using different modal data (on dataset of patients with stroke): (a) Accuracy; (b) F1-Score.

4.2.2 Regarding the Phase Recognition Task

In this subsection, the effectiveness of Basic-CNN-V1 and CNN-BiLSTM on the phase recognition task during the SitTS process for patients with stroke is further investigated in Intra-S scenario. As explained in Subsection 2.2.2, in this task, L_W is set to 32/48/64 ms, respectively, and L_O is still set to 25%×L_W, *i.e.*, 8/12/16 ms. Figs. 15(a) to 15(c) present the classification metrics of both models in five replicate experiments across all subjects at different L_W and L_O values. It can be observed that: 1) Overall, the recognition performance of both models with different modal data is slightly negatively associated with the L_W and L_O values, probably due

to the reduction in sample size; 2) Slightly different from the results in Subsection 4.2.1, both models perform better using ACC data than sEMG data regardless of the variation of L_W and L_O values, probably because the phase label segmentation is performed based on kinematic data; 3) Both models achieve the highest phase recognition accuracy exceeding 99% when using sEMG-ACC data, slightly higher than that of using sEMG-GON data.

Experimental results indicate that the proposed LLMR frameworks are also applicable to the phase recognition task during the SitTS motion process that require a lower processing delay. Additionally, for the gait-related phase recognition research driven by multimodal and multi-sensor data, they may also offer some reference.

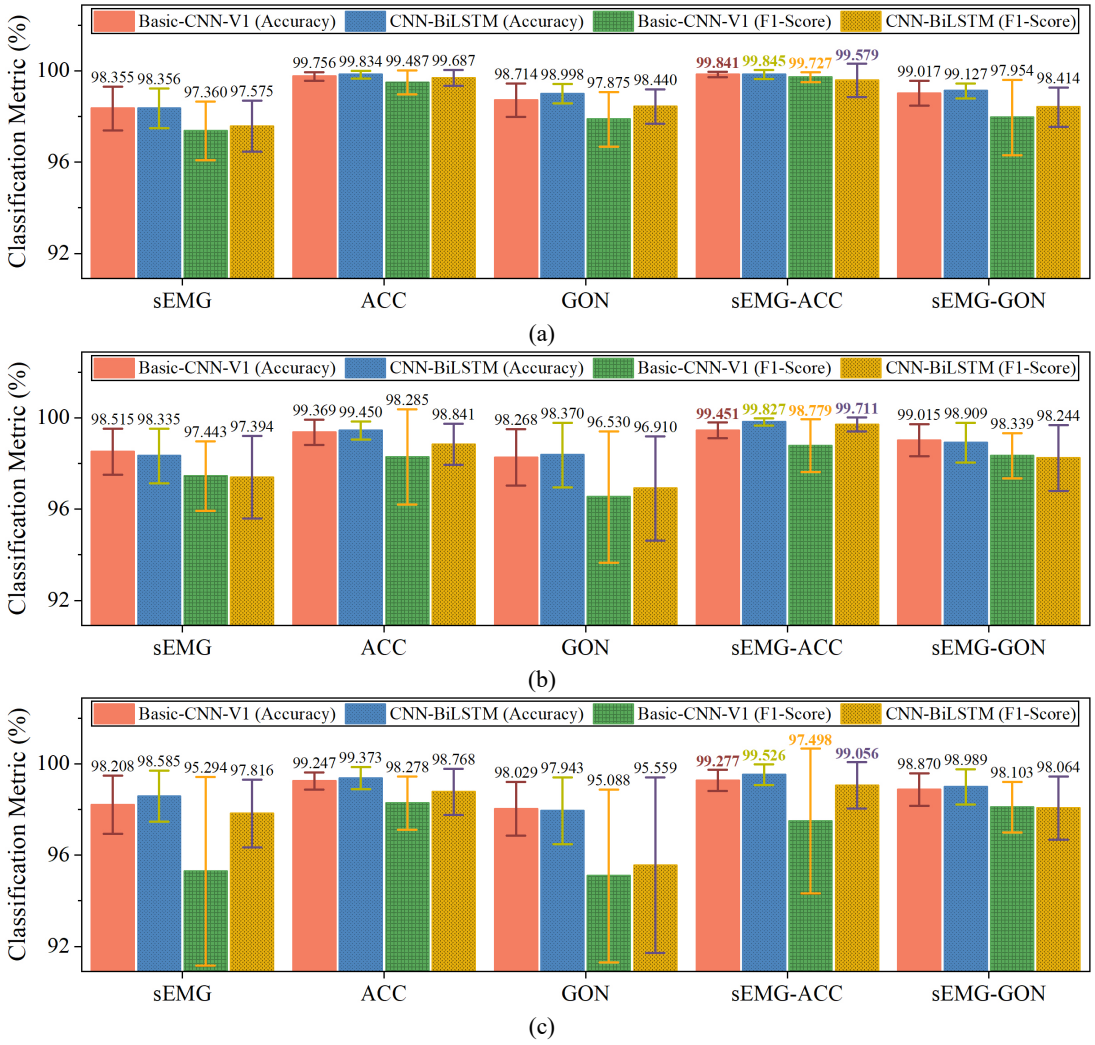


Figure 15. Variations in means and SDs of different classification performance metrics with different L_W-L_O (ms) values set across all tests of all subjects (on dataset of patients with stroke): (a) 32-8; (b) 48-12; (c) 64-16.

4.2.3 Comparison with State-of-the-Art Methods

In this section, the proposed frameworks are compared with the state-of-the-art methods mentioned in Case I. For the LLMR task, the unimodal sEMG data are uniformly used, with L_W and L_O set to 256 ms and 64 ms respectively. For the phase recognition task, the sEMG-ACC data are uniformly used, with L_W and L_O set to 32 ms and 8 ms respectively. Additionally, the relevant settings for feature extraction and hyperparameter optimization of the shallow machine

learning classifiers are described in Section 4.1.4. Experimental results are shown in Table 6.

It can be observed from Table 6 that: 1) Among the shallow machine learning classifiers, SVM performs the best on the two tasks, while among DL models, MSC-CNN performs the best; 2) For the LLMR task, the performance of shallow machine learning classifiers is close to that of the proposed frameworks, while their performance on the phase recognition task is not ideal. Experimental results further verify the superiority of the proposed frameworks in patients with stroke-related LLMR or motion phase recognition tasks.

Table 6. Comparison with state-of-the-art methods on dataset of patients with stroke.

Type	**Method**	**Accuracy (%) (Mean ± 1 SD)**	
		LLMR Task	**Phase Recognition Task**
Shallow Machine Learning	KNN [10, 12, 17]	99.127±0.761	97.212±1.491
	LDA [12, 20, 23, 54]	99.247±0.651	96.160±2.199
	SVM [10, 12, 17, 54]	**99.482±0.418**	**97.739±1.827**
	RF [21]	98.934±1.017	96.559±1.765
Deep Learning	CNN2DLSTM [55]	98.973±0.928	99.412±0.390
	CNN2DGRU [55]	99.092±0.921	99.622±0.474
	MSC-CNN [56]	**99.222±0.709**	**99.659±0.318**
Proposed	Basic-CNN-V1	99.559±0.525	99.841±0.123
	CNN-BiLSTM	**99.784±0.241**	**99.845±0.197**

5. Conclusion and Future Work

In this study, several DL-based end-to-end LLMR frameworks are explored to achieve accurate LLMR via dynamically fusing multimodal and multi-sensor information. Specially, aiming at the defect of lacking inherent graph topology in multimodal signals, a novel master nodes-based graph data generation method is presented within the CNN-GNNs framework, enabling cross-modal information flow and fusion. Furthermore, incorporating the model fine-tuning technique can significantly improve the domain generalization ability and recognition performance of the model in the Inter-S scenario at an acceptable time cost. Experimental results on two datasets demonstrate the outstanding performance of the proposed frameworks in both Intra-S and Inter-S LLMR tasks, for both healthy subjects and patients with stroke. Simultaneously, validation of effectiveness is also conducted on the phase recognition task during the SitTS process in patients with stroke. In terms of practical applications in lower limb exoskeletons or rehabilitation robots involving human-robot collaboration, requirements such as recognition accuracy, hardware performance, and computational efficiency need to be considered comprehensively to select the most suitable LLMR framework or model.

Despite the promising results of this pilot study, there are still some limitations that warrant further investigation, as described below:

1) Current work only performed offline tests, and further online evaluation is needed. Since the development of hardware resources and lightweight networks, deploying the proposed LLMR frameworks on resource-constrained embedded devices is promising as long as the sampling frequency meets the requirements for online control.

2) It is necessary to further investigate the applicability of the proposed frameworks in predicting continuous lower limb motion intentions, such as joint torques. Additionally, recognizing gait phases of healthy subjects and patients with stroke, and analyzing the differences in their gait patterns, is also one of the future works.

3) Event detection of the SitTS motion is based only on joint angle data of the affected limb in patients with stroke. However, the onset time of different events in this process is slightly different between the affected and healthy limbs. In the following work, more in-depth research will be carried out to obtain more reasonable event detection results.

Acknowledgement:

The work here is supported by the Hubei Provincial Major Science and Technology Special Project (2023BCA002). The authors are also grateful to the Editor and all Reviewers for their valuable and constructive comments which have led to a significant improvement of this work.

References:

[1] Zheng Y, Wang Y, Liu J. Analysis and experimental research on stability characteristics of squatting posture of wearable lower limb exoskeleton robot[J]. *Future Generation Computer Systems*, 2021, 125: 352-363. https://doi.org/10.1016/j.future.2021.06.053.

[2] Chen W, Lyu M, Ding X, et al. Electromyography-controlled lower extremity exoskeleton to provide wearers flexibility in walking[J]. *Biomedical Signal Processing and Control*, 2023, 79: 104096. https://doi.org/10.1016/j.bspc.2022.104096.

[3] Wu Q, Chen Y. Adaptive cooperative control of a soft elbow rehabilitation exoskeleton based on improved joint torque estimation[J]. *Mechanical Systems and Signal Processing*, 2023, 184: 109748. https://doi.org/10.1016/j.ymssp.2022.109748.

[4] Wang J, Liu J, Zhang G, et al. Periodic event-triggered sliding mode control for lower limb exoskeleton based on human–robot cooperation[J]. *ISA transactions*, 2022, 123: 87-97. https://doi.org/10.1016/j.isatra.2021.05.039.

[5] Islam M M, Nooruddin S, Karray F, et al. Multi-level feature fusion for multimodal human activity recognition in Internet of Healthcare Things[J]. *Information Fusion*, 2023, 94: 17-31. https://doi.org/10.1016/j.inffus.2023.01.015.

[6] Feng Y, Wang H, Vladareanu L, et al. New motion intention acquisition method of lower limb rehabilitation robot based on static torque sensors[J]. *Sensors*, 2019, 19(15): 3439. https://doi.org/10.3390/s19153439.

[7] Khodabandelou G, Moon H, Amirat Y, et al. A fuzzy convolutional attention-based GRU network for human activity recognition[J]. *Engineering Applications of Artificial Intelligence*, 2023, 118: 105702. https://doi.org/10.1016/j.engappai.2022.105702.

[8] García-de-Villa S, Casillas-Pérez D, Jimenez-Martin A, et al. Simultaneous exercise recognition and evaluation in prescribed routines: Approach to virtual coaches[J]. *Expert Systems with Applications*, 2022, 199: 116990. https://doi.org/10.1016/j.eswa.2022.116990.

[9] Gu L, Jiang J, Han H, et al. Recognition of unilateral lower limb movement based on EEG signals with ERP-PCA analysis[J]. *Neuroscience Letters*, 2023: 137133. https://doi.org/10.1016/j.neulet.2023.137133.

[10] Wei C, Wang H, Lu Y, et al. Recognition of lower limb movements using empirical mode decomposition and k-nearest neighbor entropy estimator with surface electromyogram signals[J]. *Biomedical Signal Processing and Control*, 2022, 71: 103198. https://doi.org/10.1016/j.bspc.2021.103198.

[11] Vijayvargiya A, Khimraj, Kumar R, et al. Voting-based 1D CNN model for human lower limb activity recognition using sEMG signal[J]. *Physical and Engineering Sciences in Medicine*, 2021, 44(4): 1297-1309. https://doi.org/10.1007/s13246-021-01071-6.

[12] Wei C, Wang H, Hu F, et al. Single-channel surface electromyography signal classification with variational mode decomposition and entropy feature for lower limb movements recognition[J]. *Biomedical Signal Processing and Control*, 2022, 74: 103487. https://doi.org/10.1016/j.bspc.2022.103487.

[13] Gozzi N, Malandri L, Mercorio F, et al. XAI for myo-controlled prosthesis: Explaining EMG data for hand gesture classification[J]. *Knowledge-Based Systems*, 2022, 240: 108053. https://doi.org/10.1016/j.knosys.2021.108053.

[14] Hooda N, Das R, Kumar N. Fusion of EEG and EMG signals for classification of unilateral foot movements[J]. *Biomedical Signal Processing and Control*, 2020, 60: 101990. https://doi.org/10.1016/j.bspc.2020.101990.

[15] Al-Quraishi M S, Elamvazuthi I, Tang T B, et al. Multimodal Fusion Approach Based on EEG and EMG Signals for Lower Limb Movement Recognition[J]. *IEEE Sensors Journal*, 2021, 21(24): 27640-27650. https://doi.org/10.1109/JSEN.2021.3119074.

[16] Dong D, Ma C, Wang M, et al. A low-cost framework for the recognition of human motion gait phases and patterns based on multi-source perception fusion[J]. *Engineering Applications of Artificial Intelligence*, 2023, 120: 105886. https://doi.org/10.1016/j.engappai.2023.105886.

[17] Zhou B, Wang H, Hu F, et al. Accurate recognition of lower limb ambulation mode based on surface electromyography and motion data using machine learning[J]. *Computer Methods and Programs in Biomedicine*, 2020, 193: 105486. https://doi.org/10.1016/j.cmpb.2020.105486.

[18] Li J, Wang Q. Multi-modal bioelectrical signal fusion analysis based on different acquisition devices and scene settings: Overview, challenges, and novel orientation[J]. *Information Fusion*, 2022, 79: 229-247. https://doi.org/10.1016/j.inffus.2021.10.018.

[19] Unanyan N N, Belov A A. Design of upper limb prosthesis using real-time motion detection method based on EMG signal processing[J]. *Biomedical Signal Processing and Control*, 2021, 70: 103062. https://doi.org/10.1016/j.bspc.2021.103062.

[20] Naik G R, Selvan S E, Arjunan S P, et al. An ICA-EBM-based sEMG classifier for recognizing lower limb movements in individuals with and without knee pathology[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018, 26(3): 675-686. https://doi.org/10.1109/TNSRE.2018.2796070.

[21] Shen C, Pei Z, Chen W, et al. Lower Limb Activity Recognition using sEMG Signals via Weighted Random Forest[C]//2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA). *IEEE*, 2022: 1151-

1156. https://doi.org/10.1109/ICIEA54703.2022.10005913.

[22] Zhang P, Zhang J, Elsabbagh A. Lower Limb Motion Intention Recognition Based on sEMG Fusion Features[J]. *IEEE Sensors Journal*, 2022, 22(7): 7005-7014. https://doi.org/10.1109/JSEN.2022.3146446.

[23] Vijayvargiya A, Gupta V, Kumar R, et al. A hybrid WD-EEMD sEMG feature extraction technique for lower limb activity recognition[J]. *IEEE Sensors Journal*, 2021, 21(18): 20431-20439. https://doi.org/10.1109/JSEN.2021.3095594.

[24] Si X, Dai Y, Wang J. Recognition of Lower Limb Movements Baesd on Electromyography (EMG) Texture Maps[C]//2022 IEEE 5th International Conference on Electronics Technology (ICET). *IEEE*, 2022: 1091-1095. https://doi.org/10.1109/ICET55676.2022.9824410.

[25] Lu Y, Wang H, Qi Y, et al. Evaluation of classification performance in human lower limb jump phases of signal correlation information and LSTM models[J]. *Biomedical Signal Processing and Control*, 2021, 64: 102279. https://doi.org/10.1016/j.bspc.2020.102279.

[26] Wu X, Yuan Y, Zhang X, et al. Gait phase classification for a lower limb exoskeleton system based on a graph convolutional network model[J]. *IEEE Transactions on Industrial Electronics*, 2021, 69(5): 4999-5008. https://doi.org/10.1109/TIE.2021.3082067.

[27] Tu J, Dai Z X, Zhao X, et al. Lower limb motion recognition based on surface electromyography[J]. *Biomedical Signal Processing and Control*, 2023, 81: 104443. https://doi.org/10.1016/j.bspc.2022.104443.

[28] Yoo Y, Jo H, Ban S W. Lite and efficient deep learning model for bearing fault diagnosis using the CWRU dataset[J]. *Sensors*, 2023, 23(6): 3157. https://doi.org/10.3390/s23063157.

[29] Li Y, Wang N, Shi J, et al. Adaptive batch normalization for practical domain adaptation[J]. *Pattern Recognition*, 2018, 80: 109-117. https://doi.org/10.1016/j.patcog.2018.03.005.

[30] Gautam A, Panwar M, Biswas D, et al. MyoNet: A transfer-learning-based LRCN for lower limb movement recognition and knee joint angle prediction for remote monitoring of rehabilitation progress from sEMG[J]. *IEEE journal of translational engineering in health and medicine*, 2020, 8: 1-10. https://doi.org/10.1109/JTEHM.2020.2972523.

[31] Zhang L, Soselia D, Wang R, et al. Lower-limb joint torque prediction using LSTM neural networks and transfer learning[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2022, 30: 600-609. https://doi.org/10.1109/TNSRE.2022.3156786.

[32] Hu B, Rouse E, Hargrove L. Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals[J]. *Frontiers in Robotics and AI*, 2018, 5: 14. https://doi.org/10.3389/frobt.2018.00014.

[33] Sun N, Cao M, Chen Y, et al. Continuous Estimation of Human Knee Joint Angles by Fusing Kinematic and Myoelectric Signals[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2022, 30: 2446-2455. https://doi.org/10.1109/TNSRE.2022.3200485.

[34] Li Y A, Chen Z J, He C, et al. Exoskeleton-Assisted Sit-to-Stand Training Improves Lower-Limb Function through Modifications of Muscle Synergies in Subacute Stroke Survivors[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2023. https://doi.org/10.1109/TNSRE.2023.3297737.

[35] Norman-Gerum V, McPhee J. Comprehensive description of sit-to-stand motions using force and angle data[J]. *Journal of Biomechanics*, 2020, 112: 110046. https://doi.org/10.1016/j.jbiomech.2020.110046.

[36] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). *IEEE*, 2016: 770-778. https://doi.org/10.1109/CVPR.2016.90.

[37] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks[C]// Proceedings of the European conference on computer vision (ECCV). *Springer International Publishing*, 2016: 630-645. https://doi.org/10.1007/978-3-319-46493-0_38.

[38] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. *arXiv preprint arXiv*:1704.04861, 2017. https://doi.org/10.48550/arXiv.1704.04861.

[39] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). *IEEE*, 2018: 4510-4520. https://doi.org/10.48550/arXiv.1801.04381.

[40] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision (ICCV). *IEEE*, 2019: 1314-1324. https://doi.org/10.1109/ICCV.2019.00140.

[41] Hu J, Shen L, Sun G. Squeeze-and-excitation networks [C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). *IEEE*, 2018: 7132-7141. https://doi.org/10.1109/TPAMI.2019.2913372.

[42] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). *IEEE*, 2018: 6848-6856. https://doi.org/10.1109/CVPR.2018.00716.

[43] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). *Springer International Publishing*, 2018: 116-131. https://doi.org/10.1007/978-3-030-01264-9_8.

[44] Xiong D, Zhang D, Zhao X, et al. Deep learning for EMG-based human-machine interaction: A review[J]. *IEEE/CAA Journal of Automatica Sinica*, 2021, 8(3): 512-533. https://doi.org/10.1109/JAS.2021.1003865.

[45] Zhang C, Wang X, Yu Z, et al. Interpretable Dual-branch EMGNet: A transfer learning-based network for inter-subject lower limb motion intention recognition[J]. *Engineering Applications of Artificial Intelligence*, 2024, 130: 107761. https://doi.org/10.1016/j.engappai.2023.107761.

[46] Yu Z, Zhang C, Deng C. An improved GNN using dynamic graph embedding mechanism: A novel end-to-end framework for rolling bearing fault diagnosis under variable working conditions[J]. *Mechanical Systems and Signal Processing*, 2023, 200: 110534. https://doi.org/10.1016/j.ymssp.2023.110534.

[47] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[J]. *Advances in neural information processing systems*, 2016, 29. https://dl.acm.org/doi/10.5555/3157382.3157527.

[48] Brody S, Alon U, Yahav E. How attentive are graph attention networks?[J]. arxiv preprint arxiv:2105.14491, 2021. https://doi.org/10.48550/arXiv.2105.14491.

[49] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. *Advances in neural information processing systems*, 2017, 30. https://dl.acm.org/doi/10.5555/3295222.3295349.

[50] Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). *IEEE*, 2021: 13713-13722.

https://doi.org/10.48550/arXiv.2103.02907.

[51] Yang S, Li M, Wang J. Fusing sEMG and EEG to increase the robustness of hand motion recognition using functional connectivity and GCN[J]. *IEEE Sensors Journal*, 2022, 22(24): 24309-24319. https://doi.org/10.1109/JSEN.2022.3221417.

[52] Massa S M, Riboni D, Nazarpour K. Graph Neural Networks for HD EMG-based Movement Intention Recognition: An Initial Investigation[C]//2022 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE). *IEEE*, 2022: 1-4. https://doi.org/10.1109/RASSE54974.2022.9989657.

[53] Zhang K, Wang J, de Silva C W, et al. Unsupervised cross-subject adaptation for predicting human locomotion intent[J]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2020, 28(3): 646-657. https://doi.org/10.1109/TNSRE.2020.2966749.

[54] Hu B, Rouse E, Hargrove L. Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities[J]. *Frontiers in Robotics and AI*, 2018, 5: 78. https://doi.org/10.3389/frobt.2018.00078.

[55] Lu H, Schomaker L R B, Carloni R. IMU-based deep neural networks for locomotor intention prediction[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). *IEEE*, 2020: 4134-4139. https://doi.org/10.1109/IROS45743.2020.9341649.

[56] Lu Z, Narayan A, Yu H. A deep learning based end-to-end locomotion mode detection method for lower limb wearable robot control[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). *IEEE*, 2020: 4091-4097. https://doi.org/10.1109/IROS45743.2020.9341183.