



DCSNN: An Efficient and High-speed sEMG-based Transient-state Micro-gesture Recognition Method on Wearable Devices

YOUFANG HAN, WEI ZHAO, GE GAO, XIANGJIN CHEN, JILiang YIN, LIN WANG, and XIN MENG, Goertek Inc., China

YANG YU* and TENGXIANG ZHANG*, Goertek Inc., China

Micro-gesture recognition using wearable devices is an important research topic in human-computer interaction. Surface electromyography (sEMG) is widely researched for gesture recognition due to its ability to capture muscle signals that precede actual gestures. Most existing methods are based on artificial neural networks (ANN), which may lead to high latency, high power consumption, and high memory usage when deployed on wearable devices. We propose a deep compressed spiking neural network (DCSNN) to address the challenges. The DCSNN can significantly reduce the inference power consumption and memory usage while improving recognition accuracy. In addition, we designed a linear method of action detection called leaky integrate-and-fire for transient-state action detection (TAD-LIF), which can improve the robustness of recognition systems effectively. To evaluate our method, we developed two lightweight sEMG wristbands respectively for two interaction modes, and collected two datasets from about 40 subjects. The experiment results show that DCSNN had a higher recognition accuracy than existing methods with values of 88.55% and 95.76% on the two datasets. In addition, its inference latency, power consumption, and memory usage are only about 0.4%, 0.05%, and 2% of those of popular convolutional neural network (CNN) methods. Our method enables precise, high-speed, and low-power micro-gesture recognition on a plethora of resource-constrained consumer-level intelligent wearable devices.

CCS Concepts: • Human-centered computing → Gestural input; • Networks → Network performance modeling.

Additional Key Words and Phrases: Surface Electromyography, Spiking Neural Network, Gesture Recognition, Action Detection, Wearable Device

ACM Reference Format:

Youfang Han, Wei Zhao, Ge Gao, Xiangjin Chen, Jiliang Yin, Lin Wang, Xin Meng, Yang Yu, and Tengxiang Zhang. 2025. DCSNN: An Efficient and High-speed sEMG-based Transient-state Micro-gesture Recognition Method on Wearable Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 2, Article 33 (June 2025), 34 pages. <https://doi.org/10.1145/3729494>

1 Introduction

Micro-gesture interaction [63] has gradually emerged as a prevalent mode of interaction due to its naturalness and privacy, particularly as augmented reality (AR) and virtual reality (VR) [25, 44] are gaining traction. Compared to hand poses, micro gestures are more natural with less user effort, postponing the onset of user fatigue. Gestures can be defined to control cursor movement in different directions, like sliding the thumb up/down/left/right

*Corresponding author.

Authors' Contact Information: Youfang Han, fred.hanyf@goertek.com; Wei Zhao, charles.zhaow@goertek.com; Ge Gao, gerard.gao@goertek.com; Xiangjin Chen, legend.chen@goertek.com; Jiliang Yin, giland.yin@goertek.com; Lin Wang, wolfgang.wang@goertek.com; Xin Meng, shawn.meng@goertek.com, Goertek Inc., Beijing, China; Yang Yu, levin.yu@goertek.com; Tengxiang Zhang, eric.zhangtx@goertek.com, Goertek Inc., Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 2474-9567/2025/6-ART33

<https://doi.org/10.1145/3729494>

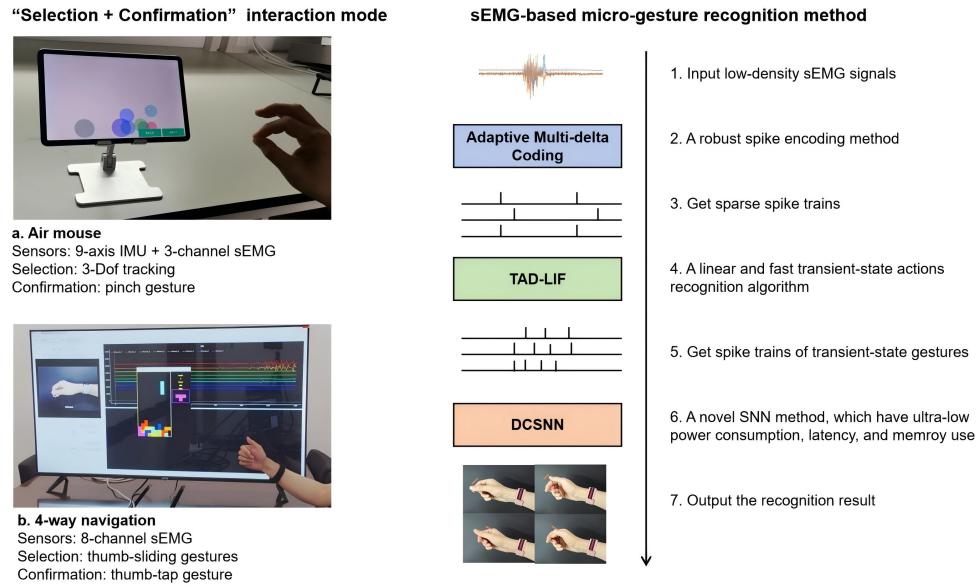


Fig. 1. The left side illustrates the application scenarios of our work, i.e. two “Selection + Confirmation” interaction modes. (a) Depicts the air mouse mode, which uses 3-channel sEMG to recognize the pinch gesture for Confirmation, while Selection is enabled by Inertial measurement unit (IMU) based 3-Dof raycasting. (b) Depicts the 4-way navigation mode, which uses 8-channel sEMG to recognize thumb-sliding gestures for Selection and thumb-tap gesture for Confirmation. The right side depicts the pipeline of our sEMG-based micro-gesture recognition method, including three novel algorithms: adaptive multi-delta coding, TAD-LIF, and DCSNN. Our method achieved high recognition accuracy while maintaining ultra-low power consumption, minimal latency, and efficient memory usage.

against the index finger, to provide basic navigation and control functions [2]. Micro-gestures also play an important role in multi-modal interaction. For instance, a simple ‘pinch’ gesture can be combined with eye-tracking technology for click and confirmation functions in VR environments [33, 79], as demonstrated by Apple’s Vision Pro. The pinch gesture can also be paired with inertial measurement unit (IMU) based positional tracking algorithms [42, 60] to function as an air mouse on devices such as personal computers (PCs) and Pads.

There are various ways to recognize micro-gestures. Camera-based gesture recognition systems that use RGB or depth cameras are sensitive to light conditions and occlusions [16, 59]. IMU-based methods [15, 45, 74] can estimate the angles of moving joints by using an accelerometer and a gyroscope to capture dynamic hand gestures. IMU sensors embedded in smart bands or watches generally excel at detecting wrist gestures and hand poses, yet they exhibit subpar performance in recognizing subtle finger movements. Acoustic-based gesture recognition systems [3, 30, 47] utilize ultrasound for remote gesture recognition. However, they are prone to interference from environmental noise. Photoplethysmographic (PPG) based methods [24, 80] detect changes in the volume of blood in the body to detect the motions of the hands and fingers. However, PPG suffers from a significant delay and necessitates a prolonged time window to capture variations in blood volume, which deteriorates user experience.

A more straightforward way to recognize gestures is by using surface Electromyography (sEMG) signals [1]. sEMG records the electrical activity associated with muscle contraction that has been widely used in biotechnology, biomechanics, and neuroscience. sEMG signals have a shorter response delay than the signals of other sensors, such as IMU and PPG, since they are generated approximately 50–100 ms before the actual action occurs [5]. Therefore, sEMG is best suitable for applications necessitating rapid responses to actions, such as gaming, driving, and precision operations. sEMG-based gesture recognition systems are not limited by factors such as lighting, occlusion, and external environment conditions, effectively addressing the limitations of other sensing methods mentioned above.

Due to the unique advantages of sEMG, there are a lot of studies on the implementation of gesture interaction systems based on sEMG signals. However, numerous issues arise in the practical application of sEMG-based gesture recognition systems. 1. Most of the existing systems rely on artificial neural networks (ANNs) [18, 20, 39, 41, 43, 65], and although notable progress has been made in optimizing ANNs for edge deployment [69, 75], further reducing computational burden remains a pressing need for resource-constrained devices such as wristbands and watches; 2. Many of the existing recognition methods employ high-density sEMG systems [51, 64, 66, 73, 76, 78], dispersed electrodes [17, 40], or large-sized electrodes [37, 55, 61], rendering them impractical for widespread application scenarios due to manufacturing difficulties and cost considerations; 3. Most existing methods have focused on hand poses rather than transient-state micro gestures [17, 46, 56, 64, 73, 77] since the signal-to-noise ratio (SNR) is much lower for micro gestures than that of hand poses.

In this paper, we propose an efficient and high-speed sEMG-based method of transient-state micro-gesture recognition based on the spiking neural network (SNN). We designed a method of event detection called leaky integrate-and-fire for transient-state action detection (TAD-LIF) to identify transient-state actions. TAD-LIF significantly reduces the amount of required computation and quickly locates the target actions. We then propose a novel deep compressed spiking neural network (DCSNN), which can accurately identify transient-state micro-gestures, and achieves orders of magnitudes of improvement in terms of power consumption, inference speed, and memory usage.

Currently, publicly available datasets such as Nina Pro [7], CSL-hdemg [4], and CapgMyo [22] focus on data of hand poses or high-density sEMG. For practicality consideration, however, we hope to use as few electrodes as possible for two different micro-gesture interaction tasks: air mouse and 4-way navigation. For air mouse, we use 3-channel sEMG for pinch gesture confirmation together with IMU-based 3-DoF selection; for 4-way navigation, we use 8-channel sEMG to recognize 5 thumb-index finger micro-gestures. For each interaction task, we developed a sEMG wristband with the number and size of electrodes tailored to the task. We collected data from 28 and 43 subjects using two wristbands respectively, and compared the performance of TAD-LIF and DCSNN with those of other small-scale models (<5 MB) on the same CPU platform. TAD-LIF showed superior recall and precision in detecting transient-state actions, and exhibited the lowest latency at low action rates. TAD-LIF is capable of substantially diminishing the interference from non-target signals, thereby alleviating the computational burden on subsequent inference models. DCSNN shows 88.55% and 95.76% recognition accuracy respectively, highest among the compared models. In addition, the inference latency, power consumption, and memory usage of DCSNN are merely about 0.4%, 0.05%, and 2% of those of convolutional neural network (CNN) methods. The results of experiments showed that the proposed method has the lowest latency, power consumption, memory usage, and could more accurately recognize the six micro-gestures considered in this study than state-of-the-art (SOTA) SNN-based methods and small-scale ANN-based methods.

The contributions of this research can be summarized as follows:

- We propose DCSNN, an SNN-based micro-gesture recognition method that achieves high inference speed, low power consumption, and low memory usage. To that end, we introduce a multi-train additive solver and a multi-step additive solver to the SNN to accelerate the inference speed and improve the robustness of the

system to temporal differences of micro-gestures. Besides, we use population coding for decision-making to improve its recognition accuracy.

- We propose adaptive multi-delta coding as a method of spiking coding to improve the stability of the network and the robustness for low SNR applications by appropriately balancing the average spike rate of the input neurons for each user's signal. We also propose a fast algorithm to detect transient-state actions called TAD-LIF for SNN-based methods. TAD-LIF is activated only when a sufficient number of spikes have been input to the network, which can filter out signals from non-transient gestures and noise, thereby alleviating the inference burden on the recognition model.
- We collected two datasets from 28 and 43 subjects, using our two self-developed wristbands respectively. The results of the experiments showed that the recognition accuracy of the proposed DCSNN was higher than most SOTA methods. Its inference latency was about 250 times lower, power consumption was about 2,000 times lower, and memory usage was about 50 times lower than those of most used CNN methods. We also invited 20 users for online testing to verify the stability of our algorithm.

2 Related Work

In this section, we reviewed research on sEMG-based methods of gesture recognition, methods of identifying action events, and the spiking neural network.

2.1 sEMG-based Methods of Gesture Recognition

Gestures usually have specific symbolic meanings in human-computer interaction (HCI). By recognizing user gestures through data collected by sensors, PCs and other smart devices can understand the user's intention. EMG signals are transmitted directly from the muscles that can be used to capture subtle movements and force-related information. EMG-based methods are capable of addressing issues such as environmental impact, high latency, and insensitivity to micro gestures that are present in other types of sensors. According to the methods by which they are obtained, EMG signals can be classified into intramuscular EMG signals (iEMG) and surface EMG signals (sEMG) [48]. iEMG signals are recorded by using needle electrodes planted inside the human muscle, but this is unsuitable for use in smart wristbands. In contrast, sEMG signals are recorded directly from the surface of the human skin, and thus have been widely used for gesture recognition because the process is safe, non-invasive, and easy to implement [35]. According to the number of electrodes used, sEMG technology can be classified into high-density sEMG (HD-sEMG) and low-density sEMG (LD-sEMG). HD-sEMG signals are collected by using a highly dense array of electrodes that can capture the distribution of muscular activity at a high resolution. However, HD-sEMG is usually costly, and cannot be integrated into smart wristbands. LD-sEMG signals are collected by using low-density electrodes which have a low cost, consume a small amount of energy for data transmission, and are easy to integrate into wristbands.

Most studies in the area have used machine learning (ML) and deep learning (DL) techniques for sEMG-based gesture recognition. ML-based methods usually calculate the handcrafted features obtained from each electrode signal and input them into ML models [11, 66]. Jiang et al. used the Linear Discriminant Analysis (LDA) on a task to identify 10 gestures and reported a recognition accuracy of 91.6% [32]. Xing et al. used the wavelet packet transform for feature extraction in a Support Vector Machine (SVM) model to identify seven gestures. The model used a four-channel sEMG and recorded an accuracy of 98.4% [70]. Atlin et al. used K Nearest Neighbors (KNN) to identify thumb gestures on a joystick and reported an accuracy of 92% [82]. These studies highlight the efficacy of machine learning in sEMG-based gesture recognition, leveraging advantages such as low computational complexity, straightforward model training, and high interpretability. However, the performance and generalization capabilities of ML-based approaches are often constrained, particularly for sEMG

signals, which are prone to signal interference and exhibit substantial inter-user variability. These limitations pose significant challenges in achieving consistent and robust recognition performance in practical applications.

Due to advances in DL technology, neural networks can now extract implicit information from raw data or handcrafted features. Most of the studies used CNN to analyze sEMG signals [1, 5, 6, 14, 18, 21, 38, 39, 41, 43, 65, 66, 76]. Triwiyanto et al. classified 10 hand motions by using two-channel sEMG signals and a CNN, and their results showed that the CNN outperformed other ML-based algorithms [65]. Ulysses et al. combined CNN with transfer learning to recognize seven gestures by using eight-channel sEMG signals. This method achieved good performance on cross-user applications [18]. Lin et al. proposed the dual-step domain adaptation network (DSDAN) to achieve unsupervised domain adaptation [41]. They tested it on HD-sEMG and LD-sEMG signals and reported good performance on tasks of cross-user gesture recognition. Besides, some works try to use recurrent neural networks (RNN) to analyze EMG signals, like Long Short-Term Memory (LSTM) [53], and Gated Recurrent Unit (GRU) [12]. In contrast to traditional machine learning, these DL-based methods have exhibited enhanced feature representation capabilities and improved robustness against diverse interference factors, thereby significantly advancing the practical implementation of sEMG-based gesture recognition. Nevertheless, the substantial computational requirements and challenges associated with edge deployment underscore the necessity for further research into reducing energy consumption and computational complexity.

2.2 Action Event Detection

Action event detection (AED) is used to check whether users have performed a target gesture. When users are in a neutral state or performing non-target actions, AED methods should exclude these corresponding signals as irrelevant and avoid the waste of energy that afflicts models of continuous inference. The occurrence of gestures always accompanies muscular activities of contraction or extension. Therefore, algorithms for AED can use the changes in sEMG signals generated by muscle activity to detect the target micro-gestures.

The SOTA algorithms for AED can be categorized into those based on visual inspection, statistical methods, ML-based methods, and threshold-based methods [62]. Visual inspection [54] by experts is accurate but complex, and is difficult to implement on wristbands. Statistical methods mainly include the approximated generalized likelihood ratio (AGLR) [23] and cumulative sum (CUSUM) [28], which are complex and slow. ML-based methods usually involve training a discriminator to detect the target action [52]. These methods have a high accuracy of detection but also a high latency and high power consumption. Threshold-based methods are the most commonly used for AED based on sEMG signals. They include methods that use a single threshold, a double threshold, and an adaptive threshold. Single threshold-based methods [29] usually compare the difference between the amplitudes of adjacent sEMG signals. If the difference is over a given threshold, the corresponding point is considered to represent the onset of the action, the entirety of which is then segmented by using a sliding window. Double threshold-based methods [23] add a second threshold to avoid false-positive detection and improve accuracy. Adaptive threshold-based methods [31] segment the given signal based on its signal-to-noise ratio (SNR) or energy, and adapt to different actions by using adjustable thresholds.

Threshold-based methods are the simplest and most effective means of AED. However, they have the following limitations [62]: 1. The setting of the threshold can easily affect the detection performance. 2. They are easily disturbed by non-target actions. 3. These methods make it difficult to identify differences in transient-state actions performed by different subjects when the duration and amplitude of their actions are different.

2.3 Spiking Neural Network

The SNN is a third-generation neural network that is based on the laws of neuromorphic computing [57]. Figure 2 compares the neurons in the ANN with those in the SNN. SNNs are event-driven methods that are active only when the neuron receives or emits spikes [8]. This characteristic enables the SNN to reduce its power

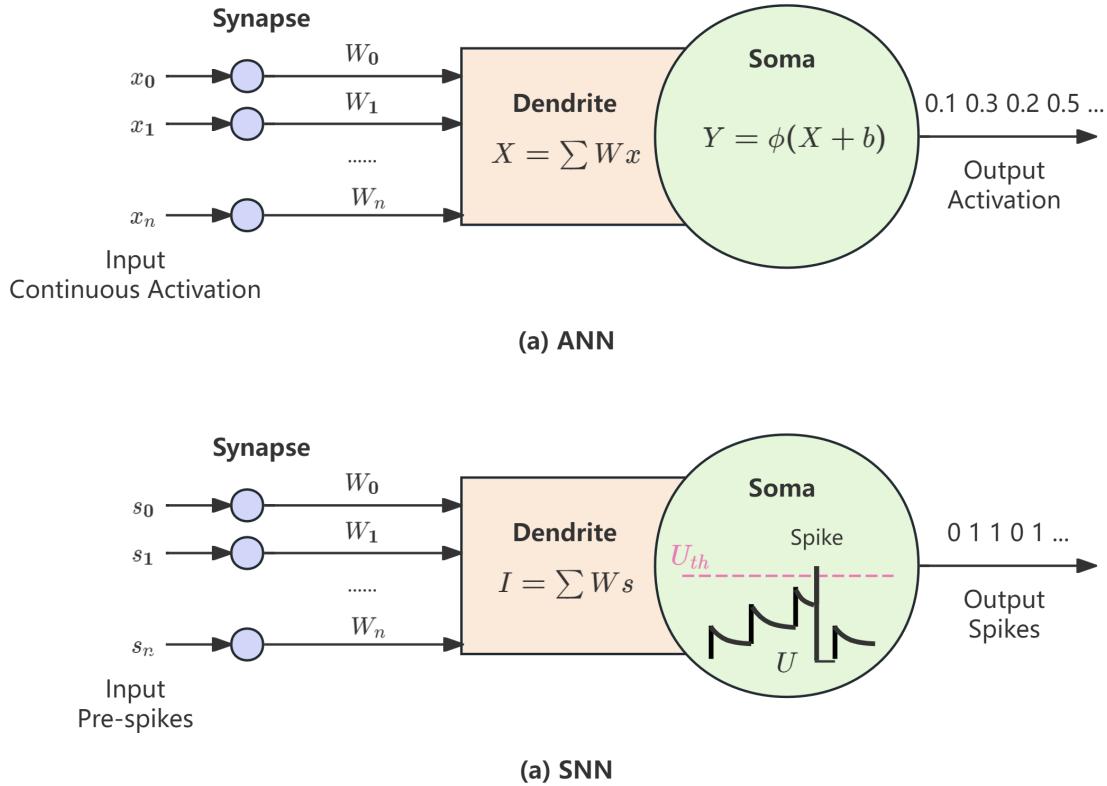


Fig. 2. Comparison between neurons of the ANN and SNN. They are different in terms of the input, output, and soma function. (a) ANN neuron. The input and output of the neuron are continuous activation values. It uses a non-linear function to process real values. (b) SNN neuron. The input and output of the neuron are spikes. It uses the procedure of leaky integrate-and-fire to process spikes.

consumption. The SNN mainly contains four parts: spike encodings, network architectures, training strategies, and hardware implementation. The spike encodings are used to convert analog and digital data into spikes [27]. The two main methods are rate coding and temporal coding [68]. The most commonly used method of spike encoding for sEMG signals is delta coding [46, 64, 73, 77], which is a type of temporal coding.

The basic unit of the SNN is the leaky integrate-and-fire (LIF) neuron. A simplified formula for LIF is shown in Eq. 1, the three parts of which correspond to leaky (potential decay), integrate (accumulation of current inputs), and fire (fire the spike). $U(t)$ represents the membrane potential at time t , β is the decay rate of the membrane potential, $X(t)$ represents the sum of spikes in each channel at time t , ε is the pre-set weight of each channel, $S(t)$ represents the output spike at time t , U_{th} represents the threshold of emitting spike.

$$U(t) = \beta U(t - 1) + \varepsilon X(t) - S(t - 1)U_{th} \quad (1)$$

In Eq.1, the value of $S(t)$ is zero or one, rendering spike trains inherently non-differentiable, and consequently necessitating distinct training strategies for SNNs compared to those employed for ANNs. The Surrogate gradient descent (SGD) methods are frequently employed to tackle this issue [34]. The SGD methods involve smoothing Dirac functions to make the data differentiable by using a surrogate function.

Besides, the SNN is more suitable than the ANN for implementation on hardware. Traditional ANNs have certain limitations in terms of deployment, and heavily rely on such computational resources as the Graphics Processing Unit (GPU) or the Neural Processing Unit (NPU). Most ANN-based models are thus deployed in the cloud. By contrast, the SNN can be implemented on neuromorphic chips, and has extremely low consumption and a high speed. With the development of neuromorphic hardware, SNNs have been widely used in object recognition [26], the analysis of biological signals [81], computer vision [71], and gesture recognition by combining bio-signals and images [13, 67].

Some recent studies have used SNN-based methods for gesture recognition based on sEMG signals. Sun et al. used the SNN on HD-sEMG and LD-sEMG signals to identify nine gestures [64]. The SNN can not only make faster inferences than alternatives, but can also solve the problems of electrode displacement and cross-user difference in sEMG. Xu et al. used the SCNN on HD-sEMG signals to identify six gestures at an accuracy of 98.78%, higher than that of a CNN with a similar structure [73]. Peng et al. used the NeuCube spiking model on a six-gesture task of recognition and reported an accuracy of 95.3% [56]. These studies have demonstrated the feasibility of SNN-based methods for sEMG-based tasks of gesture recognition. Most prevalent sEMG-based gesture recognition systems that use the SNN are used for steady-state or hand poses [17, 46, 56, 64, 73, 77]. Few studies have focused on transient-state micro-gesture recognition, but most of them have considered high-density electrode arrays [51, 64, 73] or dispersed electrodes [17], which makes it difficult to apply them to wearable devices.

3 Method

We now detailed our proposed method for transient-state micro-gesture recognition. We first introduced the method for the spike encoding of the sEMG signals, which involved transforming raw data into spike trains. We then introduced the TAD-LIF algorithm for identifying transient-state actions. Finally, we described the structure of the proposed DCSNN for gesture recognition.

3.1 Spike Encoding for sEMG

Before training the SNN, we needed to convert the raw sEMG signals into spike trains (i.e., a binary sequence of signals). Past studies [46, 64, 73, 77] have verified the effectiveness of delta coding methods for sEMG signals. We used a modified delta coding method as the method for the spike encoding of the sEMG signals. The spike encoding and pre-processing involved band-pass filtering, rectification, adaptive normalization, and adaptive multi-delta coding. To ensure better adaptation for users, we collected an item of neutral-state data and action-state data of each subject when they used the wristband for the first time.

3.1.1 Adaptive Normalization. Raw sEMG data usually contain a large amount of noise, and thus need to be filtered. We used the Butterworth band-pass filter with a bandwidth of 20–500 Hz to remove low-frequency noise and high-frequency interference from the data. We then calculated the absolute values of the data to obtain the rectified signals.

After that, the signals were regularized by using a normalization algorithm. Commonly used normalization algorithms, such as the min-max method and z-score method, can convert raw data into data within a fixed range. However, they cannot obtain the maximum, minimum, mean, variance, and other statistics in a timely manner during inference with the input signals online. Moreover, differences between the signals of users make it difficult

to set suitable statistics in advance to adapt to the data on all users. We propose an adaptive normalization method to address these issues.

We first obtained a segment of the data from the neutral state of the users as standard data when first collecting sEMG signals from them. We then calculated the median of the rectified standard data as M and normalized them by using the following formula:

$$D_{AN} = \min(1, \max(0, \frac{D_{rec} - M}{\alpha M})) \quad (2)$$

where D_{rec} represents the rectified data, D_{AN} represents the data after adaptive normalization, and α is an adjustable hyper-parameter (set to 50 in our experiments). The degree of difference between the action-state signal and the neutral-state signal can be controlled by α . If the amplitude of the user's action-state signal is closer to M , the value of α should be appropriately reduced. For specific-user, the value of α can also be directly set as the quotient of the maximum amplitude of the action-state signal divided by M . The median value of the data in the neutral state reflects the state of the sEMG signals of inaction. Slight actions by the users might have caused fluctuations in the sEMG signals when collecting the data, or external disturbances might have yielded signals with abnormally high amplitude that interfered with the results. Therefore, we used the median value of the data instead of their mean to reduce the impact of outliers.

3.1.2 Adaptive Multi-delta Coding. Delta coding is a temporal coding method that can track temporal changes in the amplitude of the signal. As shown in Eqs. 3 and 4, we obtained the spike in the signals based on the difference in their amplitudes at adjacent time points. t is the time step unit, which is same as the sEMG sampling interval. The $d(t)$ represents the amplitude of the signal at time t , θ is a threshold used to determine if the spike occurs, and $S(t)$ is the spiking function that determines whether a spike occurs at time t , and takes the value of either zero or one.

$$diff(t) = d(t) - d(t - 1) \quad (3)$$

$$S(t) = \begin{cases} 0, & \text{if } |diff(t)| < \theta \\ 1, & \text{if } |diff(t)| \geq \theta \end{cases} \quad (4)$$

However, there were usually differences in the strength of the action and the amplitude of the signal when different subjects performed the same gesture. It was difficult to use a fixed threshold θ to adapt to all gestures when using normal delta coding methods. We proposed an modified adaptive multi-delta coding method to encode the sEMG signals into a set of spike trains. Its pseudo-code is shown in Algorithm 1. Unlike existing multi-delta coding schemes [9, 10], in this algorithm, we set a maximum spike rate R_{max} to ensure that the frequency of spikes occurred over a narrow range. We collected an item of action-state data as the standard to determine the minimum threshold θ_{min} of delta coding. Following this, we used N increasing thresholds to generate N spike trains. These corresponding parameters can be set according to actual experiments conditions. For scenarios where gestures exhibit pronounced distinctions, parameter δ may be strategically increased to augment dissimilarity among spike trains, while concurrently reducing N to alleviate computational demands. Conversely, in cases involving gestures with minimal distinctions, decreasing δ while elevating N is recommended to capture more diverse signal features.

Our designed adaptive multi-delta coding calculates a user-specific minimum threshold to ensure that the maximum spike firing rates fall within a similar range across different users, preserving a reasonable level of information regardless of signal amplitude variations. By balancing the spike rates of micro-gestures across users, this method can address the issue of varying user signal strengths and low SNR, thereby improving the stability of the subsequent SNN models. We have demonstrated the effectiveness of this spike encoding method in subsequent experiments.

ALGORITHM 1: Adaptive multi-delta coding**Inputs:** sEMG signals**Initialization:** minimum threshold $\theta_{min} = 0.2$, maximum spike rate $R_{max} = 0.5$, threshold increment $\delta = 0.05$, number of spike trains $N = 10$;**If** is the **First time**: **Calculate** spike trains S by delta coding with θ_{min} , raw sEMG signals → spike trains S **While** spike rate of $S > R_{max}$: $\theta_{min}+ = \delta$; **Update** S by delta coding with θ_{min} ; $\theta = \theta_{min}$; **For** i in range(1, N): **Calculate** S_i by delta coding with θ ; $\theta+ = \delta$;**Outputs:** $\{S_1, S_2, \dots, S_N\}$

3.2 TAD-LIF

Transient-state actions can be considered to be instructions that trigger different machine commands to operate smart devices. In contrast to steady-state actions, transient-state micro-gestures have a smaller amplitude, higher speed, and lower impact on the muscles. As shown in Figure 3, a complete transient-state micro-gesture consists of a neutral-action-neutral process. This means that when performing a micro-gesture, the hand is first in a neutral state, then performs a quick action, and subsequently returns to the neutral state.

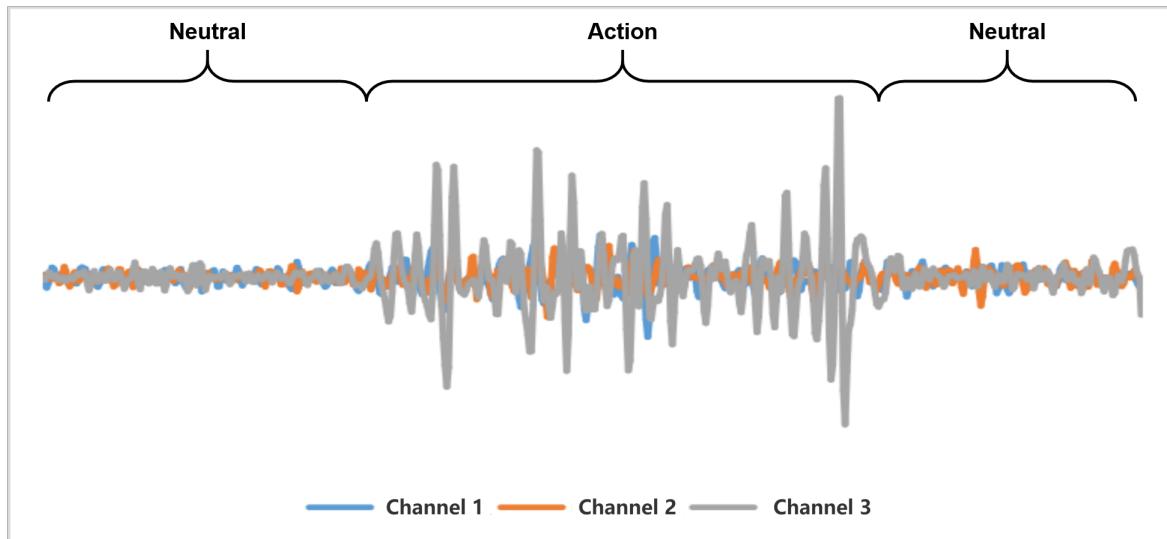


Fig. 3. An example of sEMG signals of transient-state action. It consists of three stages: a neutral stage, an action stage, and a subsequent neutral stage.

Most methods detect action segments based on digital signals and spike trains by using a sliding window with an overlap [1, 41, 52, 64, 73], or by simply inputting all segments directly to the recognition models [43]. These methods may encounter the following issues [12]: 1. The onset of actions is unclear. 2. The sliding window overlap strategy generates redundant computations. 3. The simplistic target action selection process imposes a substantial inferential burden on subsequent gesture recognition models. The goal of the proposed TAD-LIF was to address these issues and extract the signal of transient-state actions for subsequent gesture recognition.

We propose TAD-LIF to linearly detect transient-state actions without the sliding window, as shown in Figure 4. It consists of a modified LIF neuron, an action sequence buffer, and a counter. The TAD-LIF neuron linearly judges whether the given spikes are in the action state, the action sequence buffer is used to store the spike trains, and the counter records the length of the action sequence in the buffer. TAD-LIF neuron uses the following equation to update the membrane potential:

$$U_{mem}(t) = \min(\beta U_{mem}(t-1) + \omega X^2(t-1), U_{max}) \quad (5)$$

where $U_{mem}(t)$ represents the membrane potential at time t , β is the decay rate of the membrane potential, $X(t)$ represents the sum of spikes in each channel at time t , and ω is the pre-set weight of each channel. Compared with the normal LIF, TAD-LIF did not emit spikes when U_{mem} reached the threshold, nor was it reset. Instead, TAD-LIF sent a signal of the action state and added the current spike to the buffer. U_{max} was the upper limit of the U_{mem} , and prevented the potential from endlessly increasing to ensure that it was within a reasonable range. In our experiments, the weight ω was set to 0.01, the decay rate β was 0.95, and the maximum membrane potential U_{max} was 5. These hyperparameters can be configured according to specific scenarios. For instance, decreasing β can accelerate membrane potential decay, thereby preferentially selecting transient actions with higher signal amplitudes, while elevating U_{max} to enhance tolerance for low-amplitude segments within action-state signal.

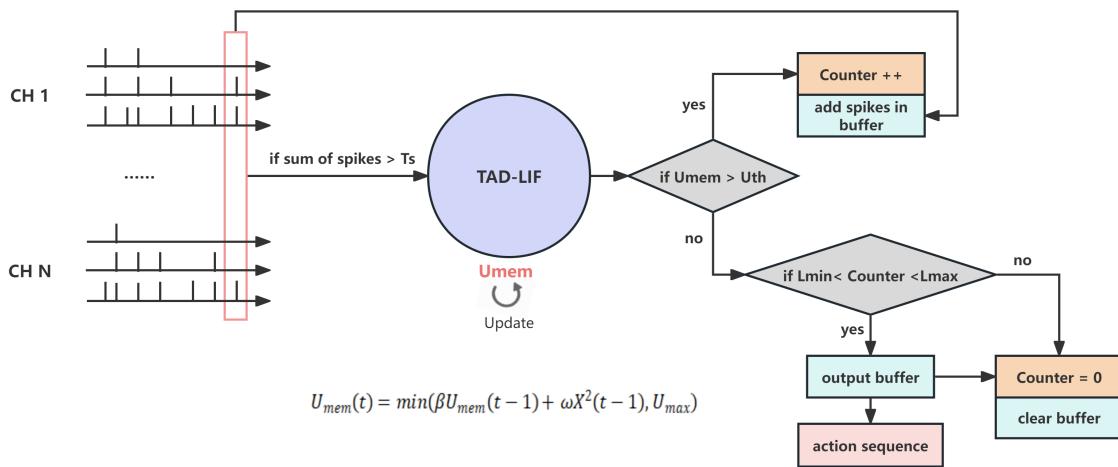


Fig. 4. Structure of TAD-LIF.

Figure 4 shows that when the sum of spikes in all channels was above the threshold T_s (set to 3), TAD-LIF was activated, and updated U_{mem} . As long as U_{mem} was higher than the threshold U_{th} (set to 0.05), the signals were considered to be in the action state. The counter was then updated and spikes were input to the buffer. The action was considered to have ended when U_{mem} was lower than U_{th} . We used L_{min} and L_{max} to limit the length of the

target action sequence, with values set to 150 and 300, respectively. These values can be adjusted based on the sampling rate and the type of action. If the length of the action sequence in the buffer satisfied the requirement of transient-state action, it was outputted to the SNN model. Otherwise, the counter and buffer were cleared directly.

To our best knowledge, TAD-LIF is the first linear AED algorithm tailored for spike trains which is compatible with SNN methods. TAD-LIF can filter out signals from non-transient gestures and noise, thereby alleviating the inference burden on the subsequent recognition model. The advantages of TAD-LIF are as follows.

- It can avoid most interference caused by irrelevant movements by controlling the membrane potential. Irrelevant movements encompass non-hand activities or basic hand translations and rotations. For such movements, the spike trains of sEMG signals from the wrist generally show minimal or no spike emission. This category of signal can be filtered by the membrane potential threshold U_{th} .
- It can filter out continuous and steady-state actions by using the counter. Continuous or steady-state finger movements typically result in ongoing spike emissions that generally persist over an extended period. We use the counter to dynamically track the duration of action signals. When the action duration exceeds acceptable thresholds for transient actions, it is subsequently filtered.
- It is a linear algorithm for detecting action sequences. TAD-LIF is not activated if no action occurs, and thus has a high detection speed and low computational cost.
- It can adapt to transient-state actions of different durations within a certain range. Because it does not have a sliding window, TAD-LIF does not have a fixed end position. It can thus adapt well to the duration of actions by different users, ranging from L_{min} to L_{max} .

3.3 DCSNN

DCSNN is an SNN-based method for sEMG-based transient-state micro-gesture recognition. The structure of the DCSNN is shown in Figure 5. A set of spike trains representing micro-gestures was input into the network. The shape of the input was $B \times C \times N \times T$, where B is the batch size ($B = 32$ in the training phase and $B = 1$ in the test phase), C is the number of channels ($C = 3$ or 8), N is the number of spike trains (set to 10, as described in Section 3.1.2), and T is the duration of the action (set to 300). The input data were processed by the two additive solvers. The multi-train additive solver added data along the dimension N . The shape of the data was transformed from $B \times C \times N \times T$ to $B \times C \times 1 \times T$. We obtained the total intensity of the spikes in each time step by using this solver. The multi-step additive solver then segmented and added the data by using a sliding window of length L (set to 20). The shape of the data was transformed from $B \times C \times 1 \times T$ to $B \times C \times 1 \times T/L$. This additive solver split the action sequence into many smaller action segments to not only reduce the size of the data, but also to improve the robustness of the model to the temporal differences between actions.

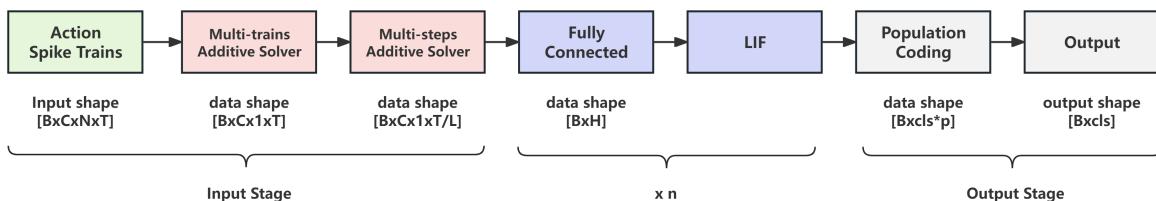


Fig. 5. Structure of the DCSNN.

Having been processed by the two additive solvers, the data were flattened to the shape $B \times H$ ($H = C \times T/L$) and then inputted to n fully connected layers and normal LIF layers. Eq. 6 shows the change in the membrane potential in the LIF neuron. $U_{mem}(t)$ represents the membrane potential at time t , β is the decay rate of the membrane potential (set to 0.5), $X(t)$ represents the spikes emitted from the pre-layers, ω is the weight of each neuron (set to 1.0), $S(t)$ represents the occurrence of spikes (zero or one) at time t , and U_{th} is the threshold used to control spike emission (set to 1.0). We use population coding in the output layer to improve the recognition accuracy of the model, as shown in Eq. 7. S_{cls} means output spikes for each class. The population coding method assigned p ($p = 100$) output neurons to each class instead of one neuron per class. The final output gesture class was determined by the summation of the p neurons of each class.

$$U_{mem}(t) = \beta U_{mem}(t - 1) + \omega X(t) - S(t - 1)U_{th} \quad (6)$$

$$\text{Output} = \text{argmax}(\sum S_{cls}) \quad (7)$$

The spiking function $S(t)$ and its derivative function are shown in Eqs. 8 and 9, respectively. $\Theta(x)$ is the Heaviside function; when $x > 0$, the result is 1, and is otherwise 0. $\delta(x)$ is the Dirac function; when $x \neq 0$, the result is 0, and is otherwise infinite.

$$S(t) = \Theta(U(t) - U_{th}); \quad \Theta(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \quad (8)$$

$$\frac{\partial S}{\partial U} = \delta(U - U_{th}); \quad \delta(x) = \begin{cases} 0, & x \neq 0 \\ \infty, & x = 0 \end{cases} \quad (9)$$

The weight of LIF cannot be updated directly by using the gradient descent method because the spiking function is the Heaviside function, the derivative $\delta(x)$ of which is non-differentiable. We used the surrogate gradient descent method to resolve this issue. Gaussian cumulative distribution function can be used as the smoothing function of the Heaviside function [67], which is differentiable. The smoothing function and its derivative are shown in Eqs. 10 and 11, respectively. $\tilde{S}(t)$ is Gaussian cumulative distribution function, $\frac{\partial \tilde{S}}{\partial U}(t)$ is Gaussian distribution function, and k is a hyper-parameter used to adjust the extent of smoothness. When calculating the gradient during training SNN, $S(t)$ will be surrogated by $\tilde{S}(t)$.

$$\tilde{S}(t) = \frac{1}{\sqrt{2\pi k}} \int_{-\infty}^x e^{-\frac{(U(t)-U_{th})^2}{2k}} dU(t) \quad (10)$$

$$\frac{\partial \tilde{S}}{\partial U}(t) = \frac{e^{-\frac{(U(t)-U_{th})^2}{2k}}}{\sqrt{2\pi k}} \quad (11)$$

4 Experiments

In this section, we start by introducing two popular micro-gesture interaction modes. Subsequently, we introduce two self-developed wristbands and datasets collected by them. We conducted offline experiments on these two datasets to verify the performance of the proposed TAD-LIF and DCSNN. Besides, we recruited some users for online experiments to test the online performance of our algorithm on wristbands. We recruited a group of users for online experiments, aiming to evaluate the end-to-end performance of our complete method (including preprocessing, TAD-LIF, and DCSNN).

The Python implementation of the proposed algorithm¹ and the complete experimental datasets² have been made publicly available for the community.

4.1 sEMG Acquisition System and Datasets

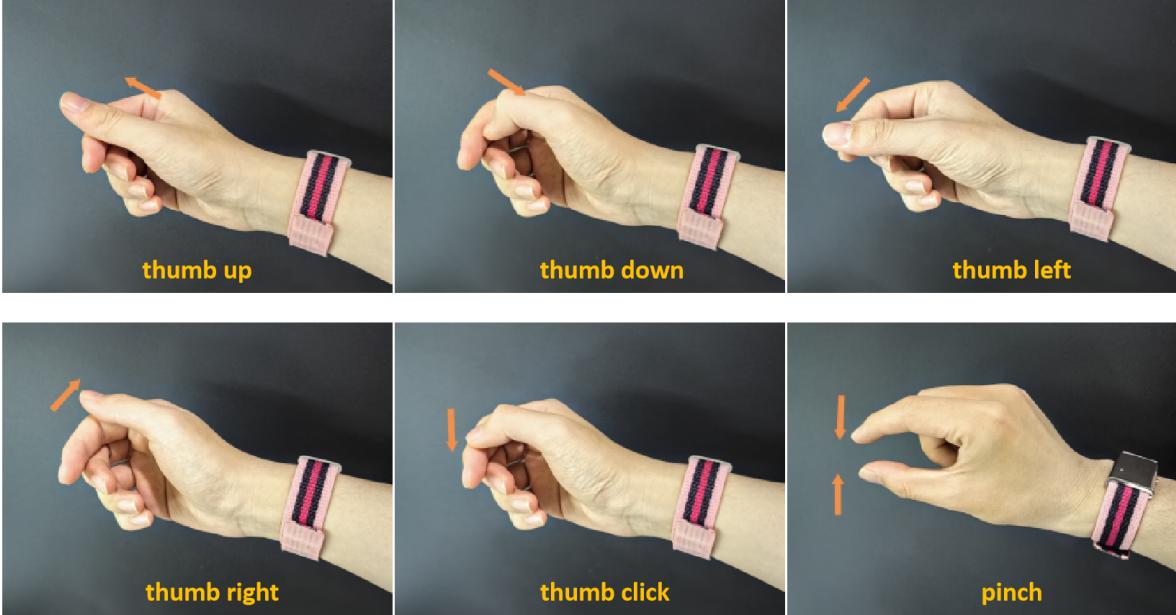


Fig. 6. Sketch of all the micro-gestures considered in this study. "Thumb up/down/left/right" means that the thumb is swiped on the index finger in different orientations. "Thumb tap" represents the tapping of the thumb on the index finger. The pinch represents quick contact between the thumb and the index finger followed by release.

4.1.1 Micro-Gesture Interaction Mode and Design of Electrode Distribution. The micro-gesture interaction mode involves users executing subtle finger movements to issue commands and interact with smart devices. The prevalent control method for smart devices is the "Selection + Confirmation" approach. The Selection operation involves navigating the cursor to a specific option, whereas the confirmation operation involves clicking and choosing this option. In this study, we focus on two "Selection + Confirmation" control approaches. The first approach employs an IMU-based 3-Dof algorithm to achieve the Selection, while the Confirmation is realized through a pinch gesture recognized by sEMG. The second approach is fully realized through micro-gestures. The Selection is achieved by swiping the index finger in different orientations, while the Confirmation is achieved by tapping of thumb on the index finger. As shown in Figure 6, we utilized sEMG to recognize 6 gestures: thumb up, thumb down, thumb left, thumb right, thumb tap, and pinch.

The necessity for sEMG electrodes to be closely adhered to the skin means that an excessive number of electrodes can lead to discomfort for users. Our work is focused on integrating a limited set of sEMG electrodes, which fulfill the criteria for micro-gesture interaction tasks, into wristbands or smartwatches. As shown in Figure 7.a, we constructed two lightweight sEMG wristbands, called Goertek Bio-Sensing (GBS) wristbands.

¹<https://github.com/Goertek-AlphaLabs-Wearable/DCSNN-for-sEMG>

²<https://www.kaggle.com/datasets/hanyoufang/dcsnn-semg-datasets>

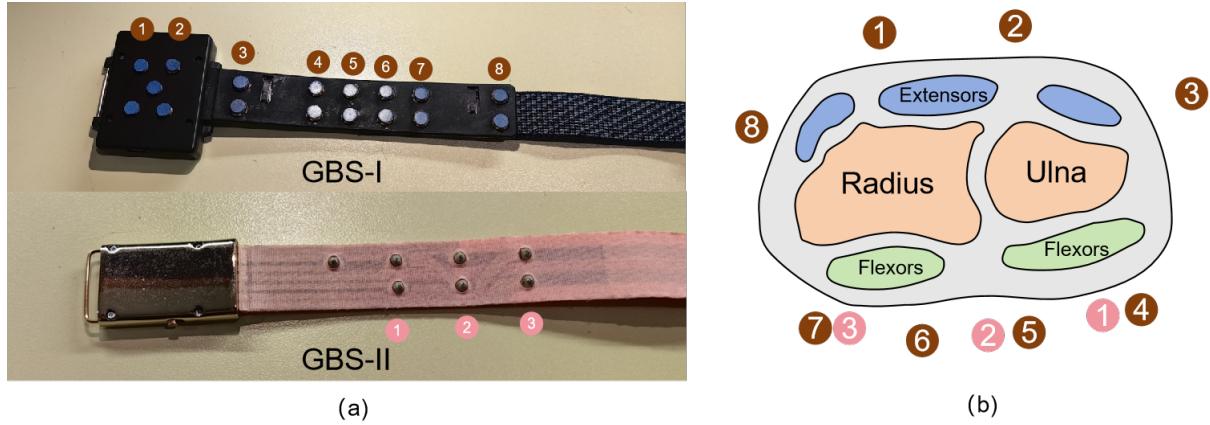


Fig. 7. Design of electrode distribution. (a) shows our self-developed Goertek Bio-Sensing (GBS) wristbands. The black band is the GBS-I wristband, which has eight electrodes, while the pink band is the GBS-II wristband, which has three electrodes. (b) is the schematic diagram illustrating the distribution of wrist muscles and bones. The brown circles indicate the placement of electrodes on the GBS-I, and the pink circles correspond to the placement of electrodes on the GBS-II.

We first referred to the electrode layout of the sEMG wristband in [38] and developed the GBS-I wristband. The distribution of the GBS-I electrodes on the wrist is depicted in Figure 7.b, where brown circles indicate the placement of four electrodes at the sites of the Flexors and Extensors, respectively. We utilized the GBS-I to collect sEMG signals of the six micro-gestures depicted in Figure 5, with the signals from each channel presented in Figure 8. We observed that five thumb-related gestures display distinct signal patterns across eight sEMG channels. To effectively differentiate these five similar micro gestures, we employed the GBS-I for thumb-related gesture recognition.

The pinch gesture consists of two distinct phases: the Hold and the Release. As depicted in the left portion of Figure 8, the signal on the left side of the red line represents the Hold phase, denoting that the thumb and index finger being pinched together. Conversely, the signal on the right side of the red line represents the Release phase, denoting that the thumb and index finger being separated. We found that the Hold is predominantly governed by the Flexors, with a clear action signal on channels 4, 5, and 6. The Release is predominantly governed by the Extensors, with an action signal on channels 1 and 2. Should the user fail to apply sufficient force during the Release, the action signal may not be obvious. Therefore, we believe that the acquisition of sEMG signals from the Flexors can effectively recognize the pinch gesture. To this end, we developed the GBS-II wristband with three electrodes placed at the sites of Flexors, as depicted by the pink circles in Figure 7. The GBS-II was integrated with an IMU sensor to enable 3-Dof tracking, thereby realizing the air mouse function.

4.1.2 sEMG Acquisition System. The existing sEMG acquisition systems [37, 55, 61] do not fulfill our requirements for a compact design with a limited number of electrodes, nor do they feature the form factor of a watch or wristband. We anticipate that the proposed algorithm will be applicable not only to independent sEMG gesture recognition systems but also to smartwatches and wristbands. Therefore, we have independently developed two GBS wristbands designed for the collection of sEMG data, with electrodes arranged in accordance with the configuration depicted in Figure 7. GBS wristbands consisted of an analog front end, an analog-to-digital converter (ADC), a battery, a Bluetooth device, an IMU, and electrodes. The GBS-I and the GBS-II contained eight/three electrodes to record sEMG signals, respectively. In Appendix A.1, we offer a comprehensive overview of the hardware specifications for GBS wristbands.

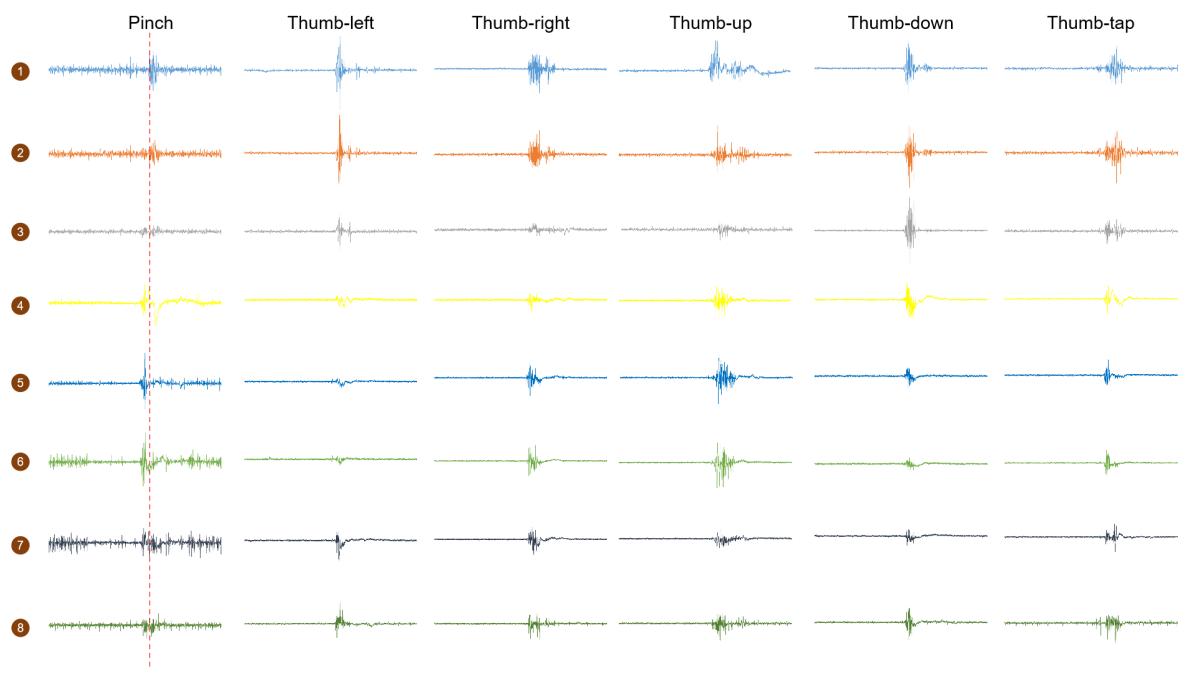


Fig. 8. Waveform of sEMG signals. The diagram illustrates the 8-channel sEMG data corresponding to 6 micro gestures, as captured by the GBS-I wristband.

Figure 9 shows a schematic diagram of the sEMG acquisition system. We used a self-developed SiP analog front end to amplify and filter the sEMG signals. BLE SOC read the results of the ADC, obtained information on the sEMG signals in real-time at a frequency of 2000Hz per channel, and sent the information to a PC in real-time.

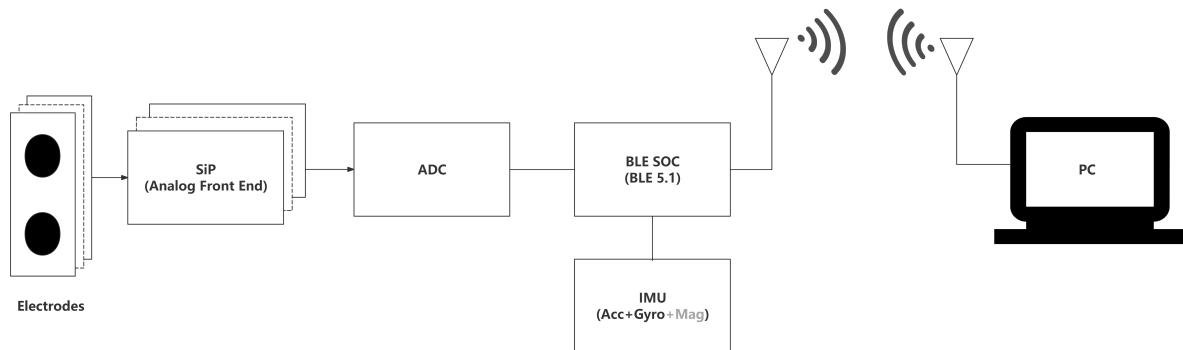


Fig. 9. Schematic diagram of sEMG acquisition system.

4.1.3 Datasets. As shown in Table 1, we used two datasets to verify the performance of the proposed methods. *Dataset I* was obtained from 28 able-bodied subjects (22 males and six females, aged 20–55 years) by the GBS-I

Table 1. Detail of the two datasets.

Dataset		Subject(#)	Gesture(#)	Repeat(#)	Electrode(#)	Sampling Rate(Hz)
<i>Dataset I</i>	training	28	6	20 - 40	8	2000
	testing	28	6	>50	8	2000
<i>Dataset II</i>	training	43	2	10 - 20	3	2000
	testing	43	2	>20	3	2000

wristband. *Dataset I* contained information on five thumb-related gestures: thumb up, thumb down, thumb left, thumb right, thumb tap, and other random gestures. *Dataset II* was obtained from 43 able-bodied subjects (37 males and six females, aged 20–55 years) by the GBS-II wristband. *Dataset II* contained information on the pinch gesture and other random gestures. When collecting random gestures, the subject can do any gesture except the pinch and thumb-related gestures, like hand open/close, wrist extension/flexion, grip, and so on. We simultaneously collect sEMG signals and IMU data. We use the IMU-based 3-Dof algorithm to implement hand tracking, and combine it with pinch gestures recognized by sEMG to achieve the air mouse function. Each subject collected two sets of data, one for model training and the other for offline testing. The interval between the two collections was about half an hour, and the subjects removed the wristbands during the interval.

The experiments on human subjects for this study were approved by the Ethics Review Board of the Wearable Products Research Center of Goertek (Beijing, China, Approval ID: GIT-HCI-2024-03-20), and informed consent was obtained from all participants.

4.2 Performance of TAD-LIF

4.2.1 Experimental Settings. We compared TAD-LIF with a non-spike threshold-based method [1, 41, 52] as well as a spike threshold-based method [64, 73]. The non-spike threshold-based method used a 300 ms window with a 50% overlap to compare the differences in the maximum and median values of the rectified signals. If the difference was greater than a fixed threshold, the signals in the windows were identified as belonging to the target segment. The spike threshold-based method set an onset when the sum of spikes at time t was above the threshold Ts. If the sum of spike trains in a 300 ms window with a 50% overlap was above a fixed threshold, these spike trains were identified as belonging to the target segment.

We tested the performance of the above methods based on recall, precision, and latency. The recall represents the ratio of correctly detected target segments, while the precision represents the ratio of the target segments to all segments in all detected actions. A high recall meant that more target segments had been found, while a high precision implied a smaller burden of inference on subsequent models. We conducted experiments using *Dataset I* and *Dataset II*, considering 6 micro-gestures as target actions, while other random gestures as non-target actions. Additionally, we added random noise into the signals of non-target action to evaluate the anti-interference ability of the algorithm.

4.2.2 Recall and Precision. The performance of the three algorithms on *Datasets I* and *II* are shown in Table 2. TAD-LIF achieves the highest recall and precision on both datasets, with its precision significantly exceeding that of the other two methods. The high recall for all three methods suggests that they are effective in detecting the target action. Precision is a measure of the method’s ability to filter out irrelevant steady-state actions, continuous actions, and noise, with a higher precision indicating a stronger capacity for anti-interference. Conversely, a low precision results in the input of a large number of irrelevant action sequences to the subsequent model, leading to a large burst of inference on it. It is evident that TAD-LIF not only sustains a very high recall to ensure the

Table 2. The recall and precision values of different methods. The optimal results on each dataset are given in **bold** while the sub-optimal results are underlined. TAD-LIF achieved the highest precision and the second-best recall on the two datasets.

	<i>Dataset I</i>		<i>Dataset II</i>	
	Recall/%	Precision/%	Recall/%	Precision/%
Non-spike Threshold	<u>99.33</u>	<u>10.24</u>	<u>99.84</u>	<u>7.89</u>
Spike Threshold	91.85	3.09	97.60	4.95
TAD-LIF	99.56	30.12	99.88	56.12

Table 3. Average latency of the three methods. The lowest latency for each action rate is given in **bold**. TAD - LIF had the lowest latency under all action rates. The experiments were implemented on a CPU with an Intel Core i7 - 6700HQ@2.60 GHz.

Action Rate	Non-spike Threshold/s	Spike Threshold/s	TAD-LIF/s
2%	3.38	2.72	1.25
10%	3.26	2.87	1.37
25%	3.25	2.93	1.49
50%	2.87	2.87	1.59
75%	2.93	3.26	1.75
100%	3.24	3.65	2.28

effective detection of micro gestures, but also markedly enhances precision, thereby alleviating the computational burden on the subsequent recognition model.

4.2.3 Latency. Spikes were emitted only when actions occurred, i.e., most of the neutral states yielded no spikes. Therefore, there was no need for action detection in most neutral states. To compare the values of delay of the three algorithms, we tested them under different rates of action. We construct sEMG sequences with different action rates from *Dataset I* and *Dataset II*. We set an action rate of 100% for the scenario in which 100 gestures occurred within 100 s. The results are shown in Table 3 and Figure 10. TAD-LIF had an average latency of about 50% of that of the threshold-based method. Figure 10 shows that the latency of the non-spike threshold-based method exhibited an irregular pattern of change, while the delay of the spike threshold-based method rose with the action rate. Therefore, TAD-LIF was superior to the other two methods, especially at a low action rate.

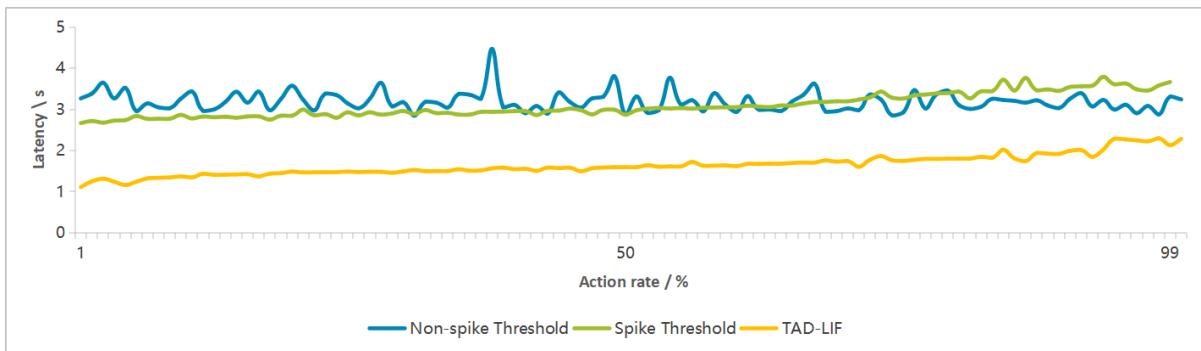


Fig. 10. Latencies of different methods under different action rates.

4.3 Performance of DCSNN

4.3.1 Experimental Settings and Comparative Models. We hope that the inference model can be deployed on edge devices, like smart watches or wristbands, so the model needs to be as small as possible to accommodate small storage hardware. Deep learning techniques are the most commonly used for gesture recognition, and many current works are based on deep learning models and have achieved good results. However, most of them were not chosen as comparative models. There are three reasons: First, some models are complicated and large-scale [6, 14, 21, 66, 76]. We expect the model size to not exceed 5 MB. Second, some works have complicated pre-processing, like spectral transformation or too much feature extraction [1, 66]. These operations will increase the computational burden of edge devices and response delay. Finally, some works focus on cross-user [18, 39, 41, 66] or few-shot tasks [50]. At present, our work only focuses on intra-user scenarios.

We choose eight representative small-scale models for comparison: CNN using one-dimensional convolutional layers (CNN-1d) [65], CNN using two-dimensional convolutional layers (CNN-2d) [82], GRU [53], SNN [64], SCNN [73], LDA [32], multi-layer perceptron (MLP) [40], MLP combining max pooling (MCMP) [49]. To verify the performance of the models fairly, we use the same training strategy, learning rate, and optimizer for ANN-based models (CNN-1d, CNN-2d, GRU, MLP, and MCMP), and use the same surrogate gradient function, learning rate, and optimizer for SNN-based models (SNN, SCNN, and DCSNN). The effectiveness of methods of machine learning is usually not as good as that of methods of neural networks [72], so we chose representative LDA [32] for comparison. In Appendix B.1, we offer a comprehensive overview of these comparative models. We used the accuracy, latency, power consumption, and memory usage as the indicators of evaluation.

4.3.2 Accuracy. We compared the accuracy of the DCSNN with those of the comparative methods on *Datasets I* and *II*. Each subject's data was used for independent experiments, and the mean and standard deviation were recorded. The results are shown in Table 4 and Figure 11. *Dataset I* contained information on five thumb-related gestures and other random gestures while *Dataset II* contained information on the pinch and other random gestures. Table 4 shows that the DCSNN obtained the accuracy of 88.55% on *Dataset I* and 95.76% on *Dataset II*. Its accuracy was considerably higher than those of the other SNN-based methods, and was even higher than that of the CNN. A one-way repeated-measure analysis of variance (ANOVA) was applied to assess the accuracy of DCSNN in comparison with other methods. The results showed that it delivered significantly better performance than other SNN-based methods. Figure 11 shows the accuracy of all methods for each gesture. The other SNN-based methods had large variance values for each gesture as well as between gestures, while DCSNN was more stable than them.

4.3.3 Inference Latency, Power Consumption, and Memory Usage. If wristbands can be used to identify the micro-gestures of users and transform them into operational instructions, users can interact more naturally with smart devices such as smartphones and computers. Therefore, the accuracy, latency, and power consumption of micro-gesture recognition systems largely determine the practicality and user experience of the corresponding devices.

We compared the inference latency, power consumption, and memory usage of different models of neural networks. The inference latency is the time required to infer one sample. All recognition models were implemented on a CPU with an Intel Core i7-6700HQ@2.60 GHz to record the average inference latency for inferring one sample. The power consumption is calculated based on the number of accumulation (AC) and multiply–accumulation (MAC) operations. We employ the method in [64, 73] for power consumption calculation, with comprehensive calculations for each model provided in Appendix B.2. The memory usage is represented by the model size.

The results shown in Table 5 were calculated on *Dataset I*. The DCSNN delivered the best performance in terms of inference latency, power consumption, and memory usage. It consumed considerably less power than DL-based methods, because ANNs required a large number of multiplication operations, while SNN-based methods

Table 4. Accuracies of different methods on data from all subjects in the two datasets. The optimal results on each dataset are given in **bold** while the sub-optimal results are underlined. The mean value is given to the left of “ \pm ” while the standard deviation is given on its right. The values in “()” show the main significant effect. One and two asterisks respectively indicate $p<0.05$ and $p<0.005$, and “ns” indicates “not significant” ($p>0.05$). Our proposed DCSNN recorded the highest accuracy on the two datasets.

Method	Dataset I/%	Dataset II/%
CNN-1d [65]	<u>87.71</u> \pm 7.11 (ns)	94.57 \pm 4.17 (ns)
CNN-2d [82]	<u>88.19</u> \pm 7.37 (ns)	93.70 \pm 4.43 (*)
GRU [53]	<u>82.60</u> \pm 9.68 (*)	94.78 \pm 5.09 (ns)
SNN [64]	<u>67.69</u> \pm 19.33 (**)	94.96 \pm 3.86 (ns)
SCNN [73]	<u>66.68</u> \pm 11.08 (**)	<u>95.05</u> \pm 6.66 (ns)
LDA [32]	<u>84.24</u> \pm 8.42 (*)	<u>85.22</u> \pm 10.56 (**)
MLP [40]	<u>83.93</u> \pm 8.35 (*)	94.83 \pm 3.88 (ns)
MCMP [49]	<u>79.98</u> \pm 11.08 (**)	91.14 \pm 9.22 (**)
DCSNN	88.55 \pm 7.89	95.76 \pm 3.39

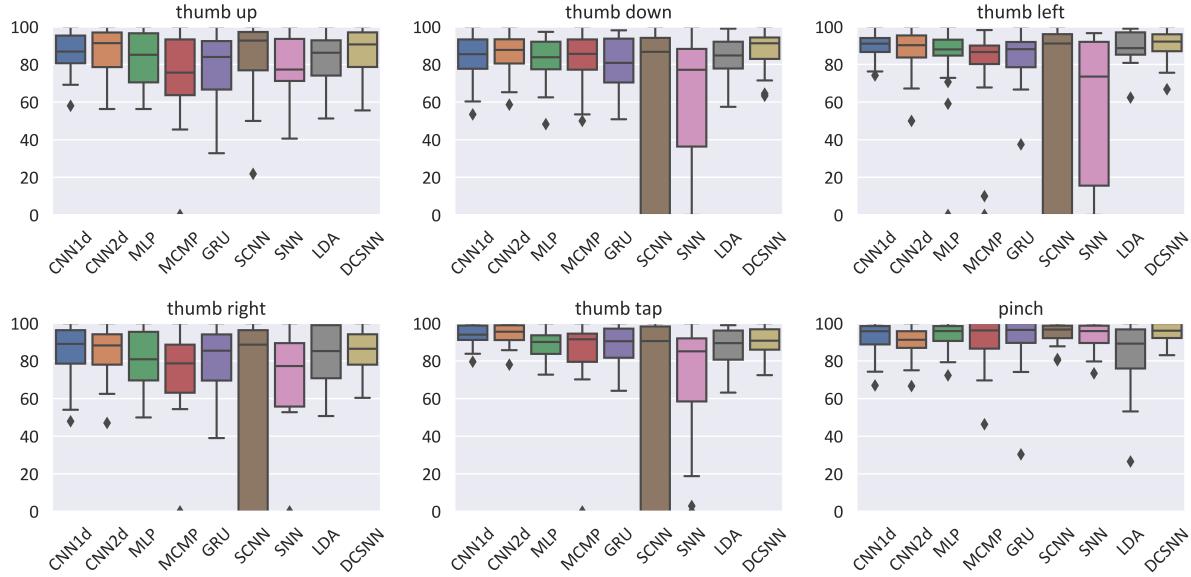


Fig. 11. Box plot of the recognition accuracy on six gestures. The results of the thumb-related gestures were tested on *Dataset I* while those of the pinch gesture were tested on *Dataset II*.

could convert many multiplication operations into addition operations. SNN-based methods used multiplication operations only for updating the membrane potential. Moreover, DCSNN contained two additive solvers before the LIF layers that further compressed the scale of the data, leading to lower power consumption and fewer parameters than the normal SNN.

4.3.4 Ablation Experiment. We verified the effectiveness of each component of DCSNN through an ablation experiment. The results are shown in Table 6 and Figure 12. The first two rows in Table 6 show the results of

Table 5. Performance of different methods. The optimal performances are given in **bold** and the sub-optimal performances are underlined. The SNNs outperformed the other methods, while our proposed DCSNN delivered the best performance.

Method	Inference latency/s	Power consumption/pJ	Model size/KB
CNN-1d	$\approx 4.5 \times 10^{-3}$	$\approx 2.5 \times 10^7$	≈ 256
CNN-2d	$\approx 8.9 \times 10^{-3}$	$\approx 7.9 \times 10^7$	≈ 4208
GRU	$\approx 1.8 \times 10^{-2}$	$\approx 1.9 \times 10^6$	≈ 772
SNN	$\approx 6.1 \times 10^{-5}$	$\approx 6.6 \times 10^5$	≈ 4938
SCNN	$\approx 3.7 \times 10^{-4}$	$\approx 9.3 \times 10^5$	≈ 4627
MLP	$\approx 4.2 \times 10^{-4}$	$\approx 2.0 \times 10^6$	≈ 5066
MCMP	$\approx 5.4 \times 10^{-5}$	$\approx 1.8 \times 10^5$	≈ 452
DCSNN	$\approx \underline{3.3} \times \mathbf{10}^{-5}$	$\approx \underline{3.8} \times \mathbf{10}^4$	≈ 84

Table 6. The results of ablation experiments. The optimal results on each dataset are given in **bold** and the sub-optimal results are underlined. The mean value is given to the left of "±" while the standard deviation is given to its right. One and two asterisks respectively indicate p<0.05 and p<0.005, and "ns" indicates "not significant" (p>0.05). Changing or removing any component of DCSNN will decrease its accuracy on the two datasets.

Method	Dataset I/%	Dataset II/%
Normal delta coding	77.87 ± 9.40 (**)	89.96 ± 9.59 (**)
Rate coding	86.50 ± 8.16 (ns)	92.21 ± 6.03 (**)
W.o. population coding	79.76 ± 12.61 (**)	92.81 ± 5.60 (*)
W.o. additive solvers	84.39 ± 8.58 (*)	92.92 ± 5.48 (*)
DCSNN	88.55 ± 7.89	95.76 ± 3.39

substituting adaptive multi-delta coding with normal delta coding and rate coding, respectively. They showed that changing the method of spike coding significantly reduced the recognition accuracy, especially when directly using normal delta coding. Without population coding or the additive solvers, the recognition accuracy of the proposed method on most gestures decreased significantly, especially on thumb-related gestures. The results of ANOVA show that changing or removing any component of DCSNN significantly degraded its performance. Therefore, adaptive multi-delta coding, population coding, and additive solvers are indispensable components of DCSNN.

4.3.5 Effect of Sampling Rate on Recognition Performance. We also investigated the impact of different signal sampling rates on recognition performance by down-sampling the raw data to 400 Hz (5× reduction), 500 Hz (4× reduction), and 1000 Hz (2× reduction). As shown in table 7, the experimental results indicate that DCSNN is more sensitive to sampling rates, and down-sampling has a greater impact on its performance. This is because DCSNN, after spike encoding and compression, suffers from the loss of original data information. For similar micro-gestures such as thumb-related actions, DCSNN relies more on the rich detail information provided by higher sampling rates to compensate for this loss. Additionally, for commonly used CNN models, reducing the sampling rate also leads to a decrease in prediction accuracy. This may be because micro-gestures are performed quickly, requiring higher sampling rates to capture subtle differences between different micro-gestures.

Based on the current experimental results, our method demonstrates significant dependence on sampling rate. For micro-gestures with higher degrees of kinematic similarity, elevated sampling rates are required to compensate for precision loss during the spike encoding process, thereby achieving satisfactory recognition outcomes.

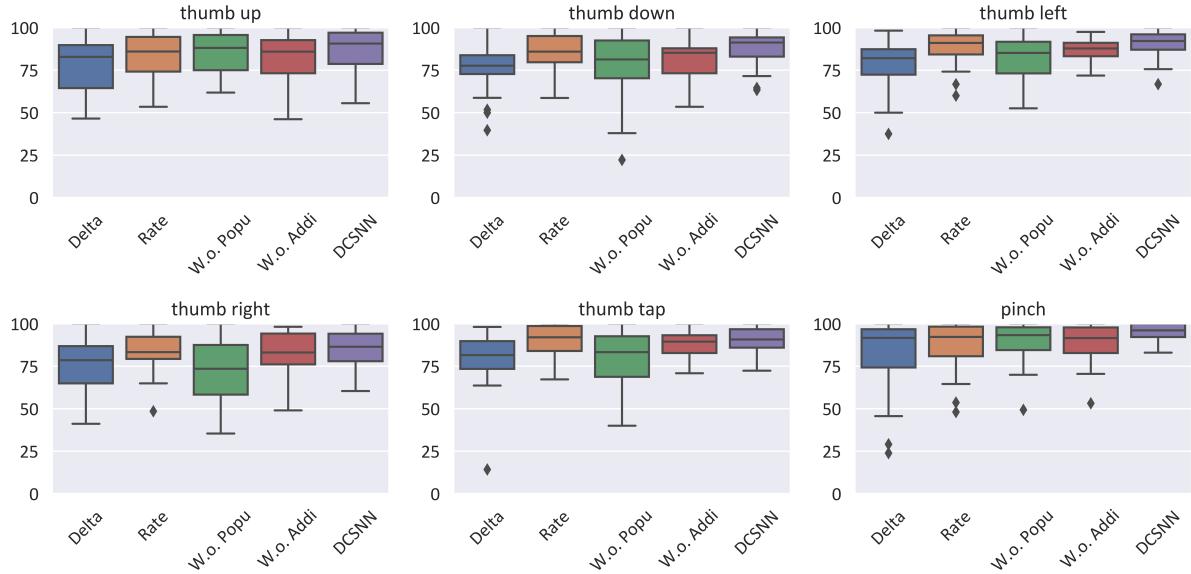


Fig. 12. Box plot of the recognition accuracies on six gestures in the ablation experiment. The methods were tested on thumb-related gestures from *Dataset I* and the pinch gestures from *Dataset II*.

Table 7. The results of the effect of sampling rate on recognition performance. The optimal results on each dataset are given in **bold** and the sub-optimal results are underlined. The mean value is given to the left of "±" while the standard deviation is given to its right. One and two asterisks respectively indicate $p < 0.05$ and $p < 0.005$, and "ns" indicates "not significant" ($p > 0.05$). Higher sampling rates generally correspond to higher performance.

Sampling	Dataset I/%		Dataset II/%	
	Rate/Hz	DCSNN	DCSNN	CNN-2d
400	65.87 ± 12.16 (**)	72.91 ± 16.39 (**)	85.64 ± 12.34 (**)	89.23 ± 9.60 (**)
500	70.99 ± 11.86 (**)	82.86 ± 11.07 (*)	89.28 ± 9.12 (*)	88.83 ± 6.50 (*)
1000	<u>79.03 ± 9.97 (*)</u>	<u>86.92 ± 8.61 (ns)</u>	<u>91.86 ± 7.25 (ns)</u>	<u>91.46 ± 7.66 (ns)</u>
2000	88.55 ± 7.89	88.19 ± 7.37	95.76 ± 3.39	93.70 ± 4.43

Consequently, when implementing the DCSNN in alternative sEMG acquisition systems, it is recommended that a sampling rate of no less than 2000 Hz be maintained to ensure optimal gesture recognition performance.

4.4 Online Evaluation

4.4.1 Evaluation Scheme. To assess the practical efficacy of our algorithm in tandem with the GBS wristband, we recruited 20 participants for an end-to-end online evaluation. Each participant dedicated roughly 10 minutes to data collection and training the gesture recognition model. Following this, they sequentially played two games, Tetris and Bubbles, as shown in Figure 13. In Tetris, the thumb-left and thumb-right gestures control the block's movement, thumb-up rotates the block, thumb-down accelerates the block's descent, and a thumb-tap pauses the game. In Bubbles, IMU-based 3-Dof controls the cursor's movement, and the pinch gesture clicks the bubbles. The

experimental process was recorded using a camera to tally the occurrences of errors. After becoming accustomed to the games, each participant conducted three experimental sessions, with a 10-minute break between each one. The detailed applications of our wristbands are shown in the supplemental video.

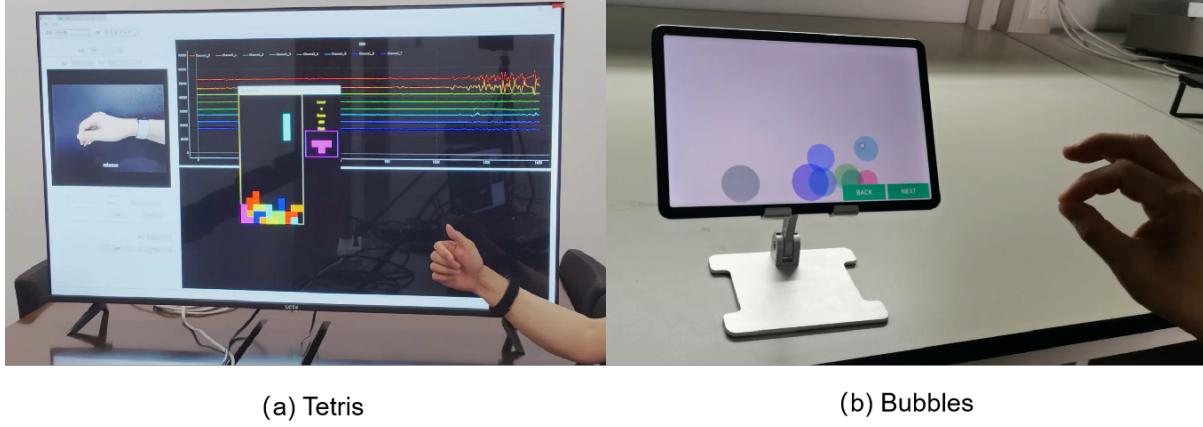


Fig. 13. Two online test games, (a) is Tetris and (b) is Bubbles.

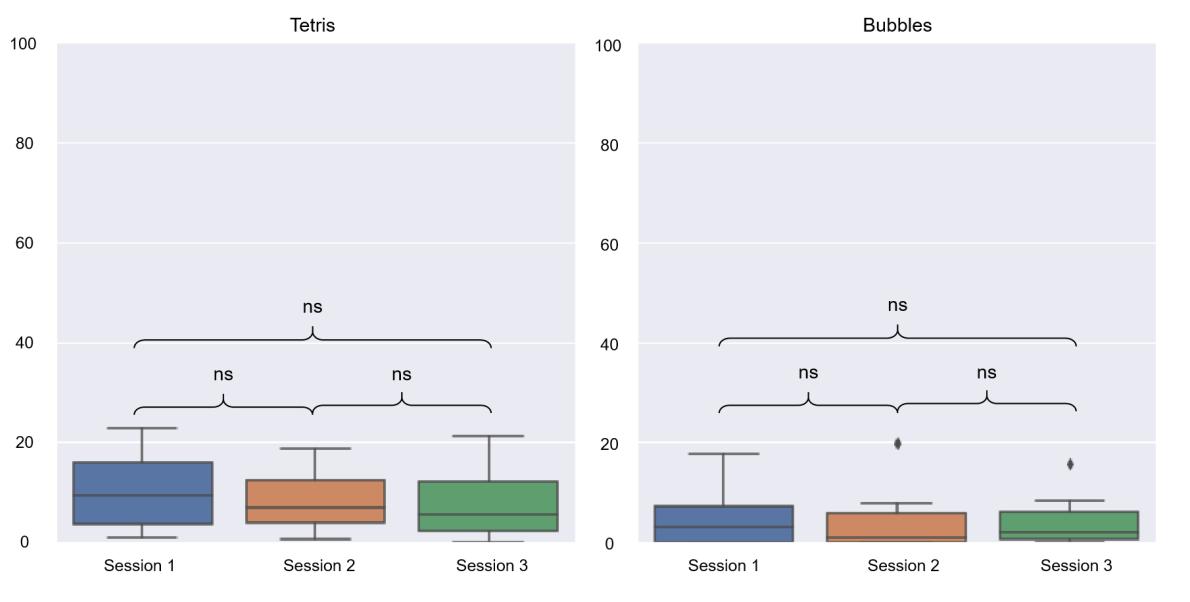


Fig. 14. Box plot of the error rate on online experiments. “ns” indicates “not significant” ($p>0.05$).

4.4.2 Online Performance. As shown in Figure 14, there are two online test games. During gameplay of Bubbles, users are prompted to perform other gestures to verify the accuracy of the pinch gesture. In each experimental

session, the number of user actions and errors are recorded. Any action that results in a mis-touch, incorrect recognition, or no recognition is classified as an error. The experimental results, as shown in Figure 14, indicate that Tetris has an average error rate of approximately 8.6% and Bubbles has an average error rate of approximately 4.1%. In Tetris, five participants with poor experience (average error rates exceeding 10% across all three sessions) believed that thumb-left and thumb-down were more prone to errors. In Bubbles, most participants noted that the pinch could be almost completely recognized, although occasionally mis-touch errors were reported, primarily by four participants (average error rates exceeding 5% across all three sessions). The results of the three sessions of experiments for each user revealed few significant differences, suggesting that this interactive control method enables users to adapt swiftly and provides a consistently positive and stable user experience.

5 Discussions

ANN-based methods have achieved remarkable accuracy in sEMG-based gesture recognition on wristband devices. However, they struggle to effectively balance high latency, high power consumption, and high memory usage, posing significant challenges for edge deployment. To address these limitations, we propose a SNN-based approach for transient micro-gesture recognition, which not only achieves superior performance but also demonstrates significant advantages in computational efficiency and energy consumption. This makes it highly suitable for deployment on edge devices such as smartwatches and wristbands. Our work represents a meaningful step toward advancing the practical application of neural-input-based human-computer interaction, potentially enabling more natural and seamless interactions in wearable ecosystems. In this section, we discuss the design principles of micro-gesture interaction, practical deployment considerations, as well as limitations and future works.

5.1 Practicality of Micro-gesture Interaction

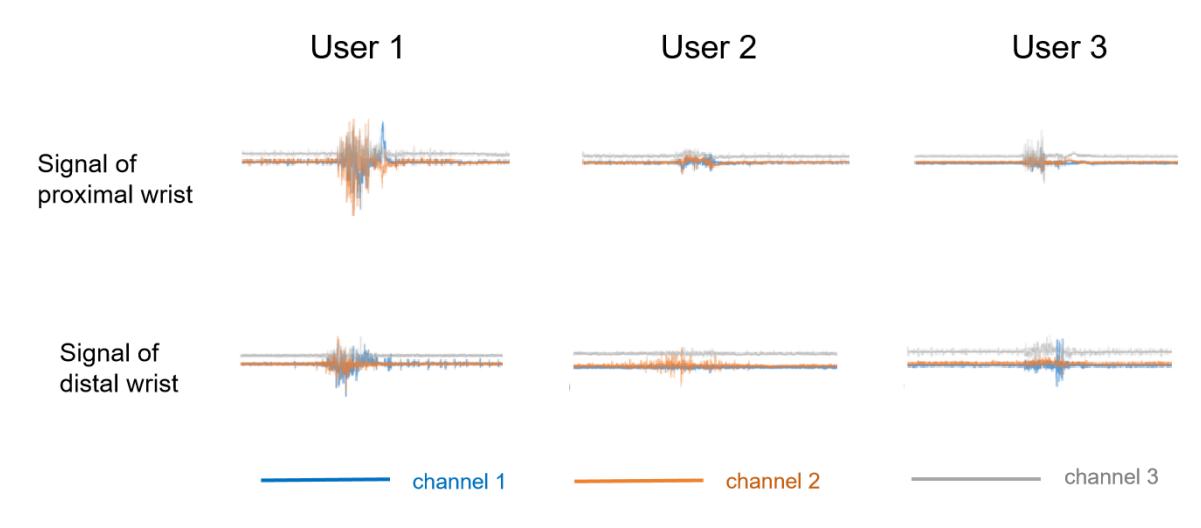


Fig. 15. Waveform of sEMG signals of proximal and distal wrist. The position of the proximal wrist is similar to the location of wearing the smartwatch. The position of the distal wrist is similar to the location of wearing Myo armbands.

The majority of sEMG control devices currently available in the market, including brands like Myo armband [37], Oymotion armband [55], and Sichiray armband [61], necessitate placement on the upper forearm to capture

sEMG signals. This requirement is at odds with the typical position for wearing watches and smart bands. In addition, most of these control devices are used to recognize hand poses, which can easily cause user fatigue. Therefore, we provide a brief analysis of the distinctions observed in sEMG signals obtained from different wearing positions and fatigue states.

We have collected sEMG signals of different wearing positions from some participants, wearing the GBS wristband on the distal wrist and proximal wrist. As shown in Figure 15, for micro-gestures, the distinction between the action and resting segments of some users is more significant when the wristband is positioned on the proximal wrist. We speculate that this phenomenon may be attributed to the subtle muscle movements associated with micro-gestures. As the thickness of the fat on the distal wrist increases, the signals of micro gesture detected by the surface electrodes tend to become less clear. The more significant the signal difference between action segments and resting segments, the more conducive it is for the DCSNN to recognize gestures. So, for micro-gesture interaction tasks, it is recommended to wear the wristband in close proximity to the wrist, aligning with the preferred location for most users when wearing smart bands and watches.

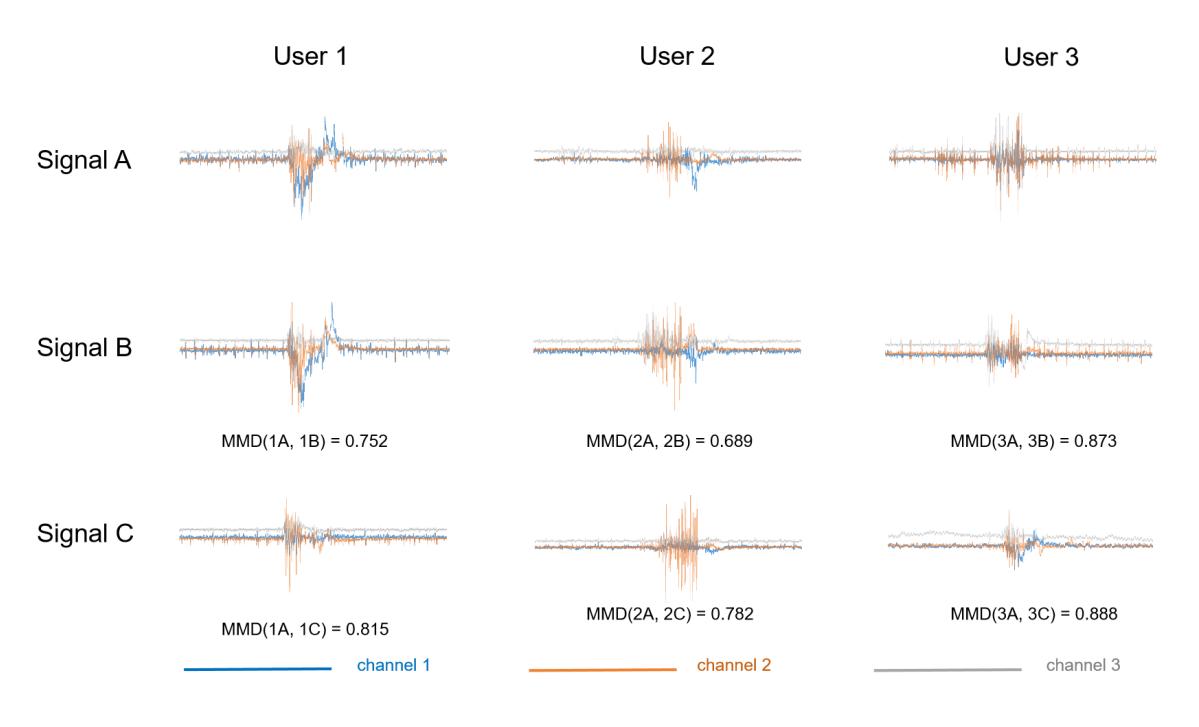


Fig. 16. Waveform of sEMG signals in Tetris fatigue experiment.

Then, we briefly analyzed the impact of fatigue on sEMG signals. We have collected sEMG signals from users who playing online games for a long time. Figure 16 illustrates three sets of sEMG signals, Signal A was recorded during the model training phase, Signal B was recorded at the beginning of online gameplay, and Signal C was recorded when the user felt fatigued. It is evident that when the user feels fatigued, the sEMG signals within the action segment become weakened in certain channels, exemplified by channel 1 in Figure 16. The Maximum Mean Discrepancy (MMD) was utilized to quantify the distributional differences between these signals. For some

users, notably User 1 and User 2, the MMD of Signal A and Signal B is significantly lower than the MMD of Signal A and Signal C. This indicates that domain shifts may occur in the sEMG signals for the same gesture when the user feels fatigued, potentially causing a reduction in gesture recognition accuracy and, as a result, a less satisfactory user experience. Therefore, when contrasted with the sEMG-based hand pose recognition tasks, the micro-gesture interaction mode boasts an inherent advantage: the subtle hand movements are less prone to fatigue. This characteristic enables users to engage in sustained interaction tasks over extended periods with greater stability.

5.2 Application Scenarios and Hardware Deployment

In this study, we utilize the proposed micro-gesture recognition method in two interaction scenarios, air mouse and 4-way navigation, to operate PCs and Pads for gaming purposes. These scenarios were chosen because they closely mimic real-world applications, providing a practical evaluation of the method's usability and effectiveness. Beyond game control, this method is also applicable to various micro-gesture interaction scenarios, including VR/AR control, smart home control, drone operation, and intelligent driving systems, among others. We verified the cross-session performance of the proposed method through offline experiments, demonstrating its robustness to variations in signal characteristics over time. This robustness is critical for real-world deployment, where users may perform gestures under different conditions or after extended periods of device usage. Additionally, the intra-session performance was validated through online experiments, achieving high recognition accuracy and further confirming the method's reliability in real-time applications. Furthermore, to ensure a fair comparison of performance between SNN-based and ANN-based methods, we conducted experiments on a consolidated CPU platform. Our proposed DCSNN, which shares a similar inference process with most SNN methods, can be deployed on widely available CPU-based chips, such as Intel Core i7 and AMD Ryzen 5.

5.3 Limitations and Future Work

Although the proposed method has made significant progress, there are still some limitations that need to be addressed. First, the gesture recognition performance of the proposed method still has room for improvement, particularly in distinguishing subtle thumb-based movements (*Dataset I*). On the one hand, the resolution of low-density sEMG (LD-sEMG) is lower than that of high-density sEMG (HD-sEMG), resulting in insufficient feature extraction. On the other hand, to meet the requirements of lower computational burden, the proposed SNN has made structural compromises, which may lead to the loss of some deeper implicit features. Second, our method currently only considers cross-session scenarios and does not address cross-user variability. The significant differences between users remain one of the most challenging issues in sEMG-based gesture recognition. Although some researchers have claimed that SNNs can alleviate the impact of electrode shifts and cross-user differences to some extent [64, 73], these experiments were conducted using high-density electrode arrays, and their effectiveness on wristband electrodes remains to be verified. Finally, our experiments were conducted on CPU and NPU-based chips, and we did not test our method on neuromorphic hardware. As a result, the inherent hardware advantages of SNNs, such as low power consumption and high efficiency, have not been fully utilized.

Based on the above-mentioned limitations, in the future, we will continue to optimize our algorithms and try to achieve higher accuracy in micro-gesture recognition while ensuring its advantages of low power consumption and low latency. In addition, we will collect more sEMG data on other micro-gestures to verify the generalization capability of our method and broaden its range of application scenarios. We will also investigate ways to combine the proposed method with transfer learning methods to address the issue of cross-user variability. Furthermore, inspired by recent advances in hand pose estimation, such as EITPose [36] and EMG2Pose [58], we plan to improve the hardware and algorithm, as well as collect more data, to further explore the potential of our method for continuous hand pose estimation tasks. Finally, we will develop or identify neuromorphic chips suitable for

integration into wearable devices, while adapting our algorithms of TAD-LIF and DCSNN. This will facilitate the progression of our techniques towards practical, real-world applications.

6 Conclusion

In this study, our proposed method incorporates two novel algorithms, TAD-LIF and DCSNN, and verified their performance against SOTA methods. We utilized a self-developed sEMG acquisition system to collect the LD-SEMG signals during our experiments. Firstly, TAD-LIF is a variant of the LIF neuron, which is designed for the purpose of identifying transient-state gestures within spike trains. From the experimental results, our proposed TAD-LIF exhibits not only lower latency, but also significantly superior precision compared to the two other most commonly used AED methods. This indicates that TAD-LIF can effectively reduce the interference of irrelevant steady-state gestures, continuous actions, noise, and other extraneous signals, and reduce the computational burden of subsequent inference models.

Furthermore, to tackle the challenge of restricted computational resources in edge devices, we proposed an SNN-based method in this study. This method was designed to be power-efficient, low inference latency, and achieve high accuracy of micro-gesture recognition. The experimental results showed that the proposed DCSNN outperformed the most-used CNN in terms of the recognition accuracy of most micro-gestures. The inference latency of DCSNN was about 0.4%, its power consumption was 0.05%, and its memory usage was 2% of those of the CNN. Our results suggested that DCSNN achieved superior recognition accuracy and exhibited the lowest resource consumption compared to the lightweight CNN models, rendering it more suitable for deployment in widely used Micro-controller Unit (MCU) like nRF52840 and Apollo 4. Furthermore, we validated the effectiveness of each component of DCSNN through ablation experiments, and the practicality of our algorithms through online experiments. These experiments show the advantages of our methods in terms of power consumption, latency, and recognition accuracy. Our proposed DCSNN and TAD-LIF effectively facilitate the application of wearable devices, such as smartwatches and smart wristbands, in sEMG-based micro-gesture recognition tasks, promoting a more natural mode of human-computer interaction for users.

Acknowledgments

This work was supported by Alpha Labs of Goertek and Beijing Goertek Technology Co.,Ltd. We thank the Wearable Products Research Center of Goertek for providing the datasets of micro-gestures analyzed in this paper.

References

- [1] Ulysse Côté Allard, François Nougarou, Cheikh Latyr Fall, Philippe Giguère, Clément Gosselin, François Laviolette, and Benoit Gosselin. 2016. A convolutional neural network for robotic arm guidance using sEMG based frequency-features. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2464–2470. [doi:10.1109/IROS.2016.7759384](https://doi.org/10.1109/IROS.2016.7759384)
- [2] Cemil Altin and O Er. 2017. Designing wearable joystick and performance comparison of EMG classification methods for thumb finger gestures of joystick control. *Biomed Res India* 28 (2017), 4730–4736.
- [3] Takashi Amesaka, Hiroki Watanabe, Masanori Sugimoto, and Buntarou Shizuki. 2022. Gesture Recognition Method Using Acoustic Sensing on Usual Garment. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2 (2022), 41–1. [doi:10.1145/3534579](https://doi.org/10.1145/3534579)
- [4] Christoph Amma, Thomas Krings, Jonas Böer, and Tanja Schultz. 2015. Advancing muscle-computer interfaces with high-density electromyography. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 929–938. [doi:10.1145/2702123.2702501](https://doi.org/10.1145/2702123.2702501)
- [5] Panagiotis Artermiadis. 2012. EMG-based robot control interfaces: past, present and future. *Advances in Robotics & Automation* 1, 2 (2012), 1–3. [doi:10.4172/2168-9695.1000e107](https://doi.org/10.4172/2168-9695.1000e107)
- [6] Manfredo Atzori, Matteo Cognolato, and Henning Müller. 2016. Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in neurorobotics* 10 (2016), 9. [doi:10.3389/fnbot.2016.00009](https://doi.org/10.3389/fnbot.2016.00009)

- [7] Manfredo Atzori, Arjan Gijsberts, Simone Heynen, Anne-Gabrielle Mittaz Hager, Olivier Deriaz, Patrick Van Der Smagt, Claudio Castellini, Barbara Caputo, and Henning Müller. 2012. Building the Niapro database: A resource for the biorobotics community. In *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 1258–1265. [doi:10.1109/BioRob.2012.6290287](https://doi.org/10.1109/BioRob.2012.6290287)
- [8] Daniel Auge, Julian Hille, Etienne Mueller, and Alois Knoll. 2021. A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters* 53, 6 (2021), 4693–4710. [doi:10.1007/s11063-021-10562-2](https://doi.org/10.1007/s11063-021-10562-2)
- [9] Sizhen Bian, Elisa Donati, and Michele Magno. 2024. Evaluation of Encoding Schemes on Ubiquitous Sensor Signal for Spiking Neural Network. *IEEE Sensors Journal* (2024).
- [10] Sizhen Bian and Michele Magno. 2023. Evaluating spiking neural network on neuromorphic platform for human activity recognition. In *Proceedings of the 2023 ACM International Symposium on Wearable Computers*. 82–86. [doi:10.1145/3594738.3611369](https://doi.org/10.1145/3594738.3611369)
- [11] Evan Campbell, Angkoon Phinyomark, Ali H Al-Timemy, Rami N Khushaba, Giovanni Petri, and Erik Scheme. 2019. Differences in EMG feature space between able-bodied and amputee subjects for myoelectric control. In *2019 9th international IEEE/EMBS conference on neural engineering (NER)*. IEEE, 33–36. [doi:10.1109/NER.2019.8717161](https://doi.org/10.1109/NER.2019.8717161)
- [12] Camila R Carvalho, J Marvin Fernández, Antonio J Del-Ama, Filipe Oliveira Barroso, and Juan C Moreno. 2023. Review of electromyography onset detection methods for real-time control of robotic exoskeletons. *Journal of neuroengineering and rehabilitation* 20, 1 (2023), 141. [doi:10.1186/s12984-023-01268-8](https://doi.org/10.1186/s12984-023-01268-8)
- [13] Enea Ceolini, Charlotte Frenkel, Sumit Bam Shrestha, Gemma Taverni, Lyes Khacif, Melika Payvand, and Elisa Donati. 2020. Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing. *Frontiers in neuroscience* 14 (2020), 637. [doi:10.3389/fnins.2020.00637](https://doi.org/10.3389/fnins.2020.00637)
- [14] Jiangcheng Chen, Sheng Bi, George Zhang, and Guangzhong Cao. 2020. High-density surface EMG-based gesture recognition using a 3D convolutional neural network. *Sensors* 20, 4 (2020), 1201. [doi:10.3390/s20041201](https://doi.org/10.3390/s20041201)
- [15] Wenqiang Chen, Ziqi Wang, Pengrui Quan, Zhencan Peng, Shupei Lin, Mani Srivastava, Wojciech Matusik, and John Stankovic. 2023. Robust finger interactions with COTS smartwatches via unsupervised siamese adaptation. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14. [doi:10.1145/3586183.3606794](https://doi.org/10.1145/3586183.3606794)
- [16] Weiya Chen, Chenchen Yu, Chenyu Tu, Zehua Lyu, Jing Tang, Shiqi Ou, Yan Fu, and Zhidong Xue. 2020. A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors* 20, 4 (2020), 1074. [doi:10.3390/s20041074](https://doi.org/10.3390/s20041074)
- [17] Long Cheng, Yang Liu, Zeng-Guang Hou, Min Tan, Dajun Du, and Minrui Fei. 2019. A rapid spiking neural network approach with an application on hand gesture recognition. *IEEE Transactions on Cognitive and Developmental Systems* 13, 1 (2019), 151–161. [doi:10.1109/TCDS.2019.2918228](https://doi.org/10.1109/TCDS.2019.2918228)
- [18] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. 2019. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE transactions on neural systems and rehabilitation engineering* 27, 4 (2019), 760–771. [doi:10.1109/TNSRE.2019.2896269](https://doi.org/10.1109/TNSRE.2019.2896269)
- [19] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, François Laviolette, and Benoit Gosselin. 2019. A low-cost, wireless, 3-D-printed custom armband for sEMG hand gesture recognition. *Sensors* 19, 12 (2019), 2811. [doi:10.3390/s19122811](https://doi.org/10.3390/s19122811)
- [20] Ulysse Côté-Allard, Gabriel Gagnon-Turcotte, Angkoon Phinyomark, Kyrre Glette, Erik Scheme, François Laviolette, and Benoit Gosselin. 2021. A transferable adaptive domain adversarial neural network for virtual reality augmented EMG-based gesture recognition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), 546–555. [doi:10.1109/TNSRE.2021.3059741](https://doi.org/10.1109/TNSRE.2021.3059741)
- [21] Zhen Ding, Chifu Yang, Zhihong Tian, Chunzhi Yi, Yunsheng Fu, and Feng Jiang. 2018. sEMG-based gesture recognition with convolution neural networks. *Sustainability* 10, 6 (2018), 1865. [doi:10.3390/su10061865](https://doi.org/10.3390/su10061865)
- [22] Yu Du, Yongkang Wong, Wenguang Jin, Wentao Wei, Yu Hu, Mohan S Kankanhalli, and Weidong Geng. 2017. Semi-Supervised Learning for Surface EMG-based Gesture Recognition.. In *IJCAI*. 1624–1630. [doi:10.5555/3172077.3172113](https://doi.org/10.5555/3172077.3172113)
- [23] PH Ellaway. 1978. Cumulative sum technique and its application to the analysis of peristimulus time histograms. *Electroencephalography and clinical neurophysiology* 45, 2 (1978), 302–304. [doi:10.1016/0013-4694\(78\)90017-2](https://doi.org/10.1016/0013-4694(78)90017-2)
- [24] Chengfeng Fang, Xin Ruan, Xu Zhang, Xun Chen, and Xiang Chen. 2023. Exploration on the negative effects of sensor shifts in photoplethysmography-based gesture recognition and a solution based on transfer learning. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–11. [doi:10.1109/TIM.2023.3268469](https://doi.org/10.1109/TIM.2023.3268469)
- [25] Jacqui Fashimpaur, Amy Karlson, Tanya R Jonker, Hrvoje Benko, and Aakar Gupta. 2023. Investigating wrist deflection scrolling techniques for extended reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16. [doi:10.1145/3544548.3580870](https://doi.org/10.1145/3544548.3580870)
- [26] Yifei Feng, Shijia Geng, Jianjun Chu, Zhaoji Fu, and Shenda Hong. 2022. Building and training a deep spiking neural network for ECG classification. *Biomedical Signal Processing and Control* 77 (2022), 103749. [doi:10.1016/j.bspc.2022.103749](https://doi.org/10.1016/j.bspc.2022.103749)
- [27] Wulfraum Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press.
- [28] Marco Ghislieri, Giacinto Luigi Cerone, Marco Knaflitz, and Valentina Agostini. 2021. Long short-term memory (LSTM) recurrent neural network for muscle activity detection. *Journal of NeuroEngineering and Rehabilitation* 18 (2021), 1–15. [doi:10.1186/s12984-021-00945-w](https://doi.org/10.1186/s12984-021-00945-w)

- [29] Paul W Hodges and Bang H Bui. 1996. A comparison of computer-based methods for the determination of onset of muscle contraction using electromyography. *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control* 101, 6 (1996), 511–519. doi:[10.1016/S0921-884X\(96\)95190-5](https://doi.org/10.1016/S0921-884X(96)95190-5)
- [30] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2019. BeamBand: Hand gesture sensing with ultrasonic beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10. doi:[10.1145/3290605.3300245](https://doi.org/10.1145/3290605.3300245)
- [31] Eugene M Izhikevich. 2006. Polychronization: computation with spikes. *Neural computation* 18, 2 (2006), 245–282. doi:[10.1162/089976606775093882](https://doi.org/10.1162/089976606775093882)
- [32] Shuo Jiang, Qinghua Gao, Huaiyang Liu, and Peter B Shull. 2020. A novel, co-located EMG-FMG-sensing wearable armband for hand gesture recognition. *Sensors and Actuators A: Physical* 301 (2020), 111738. doi:[10.1016/j.sna.2019.111738](https://doi.org/10.1016/j.sna.2019.111738)
- [33] Hansol Kim, Kun Ha Suh, and Eui Chul Lee. 2017. Multi-modal user interface combining eye tracking and hand gesture recognition. *Journal on Multimodal User Interfaces* 11 (2017), 241–250. doi:[10.1007/s12193-017-0242-2](https://doi.org/10.1007/s12193-017-0242-2)
- [34] Seijoon Kim, Seongsik Park, Byunggoon Na, and Sungroh Yoon. 2020. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 11270–11277. doi:[10.1609/aaai.v34i07.6787](https://doi.org/10.1609/aaai.v34i07.6787)
- [35] Davinder Kumar and Aman Ganesh. 2022. A critical review on hand gesture recognition using semg: Challenges, application, process and techniques. In *Journal of Physics: Conference Series*, Vol. 2327. IOP Publishing, 012075. doi:[10.1088/1742-6596/2327/1/012075](https://doi.org/10.1088/1742-6596/2327/1/012075)
- [36] Alexander Kyu, Hongyu Mao, Junyi Zhu, Mayank Goel, and Karan Ahuja. 2024. Eitpose: Wearable and practical electrical impedance tomography for continuous hand pose estimation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–10. doi:[10.1145/3613904.3642663](https://doi.org/10.1145/3613904.3642663)
- [37] Thalmics Labs. [n. d.]. Myo Armband. <https://support.getmyo.com/hc/en-us>.
- [38] Ctrl labs at Reality Labs, David Sussillo, Patrick Kaifosh, and Thomas Reardon. 2024. A generic noninvasive neuromotor interface for human-computer interaction. *Biorxiv* (2024), 2024–02. doi:[10.1101/2024.02.23.581779](https://doi.org/10.1101/2024.02.23.581779)
- [39] Xinhui Li, Xu Zhang, Xiang Chen, Xun Chen, and Aiping Liu. 2023. Cross-user gesture recognition from sEMG signals using an optimal transport assisted student-teacher framework. *Computers in Biology and Medicine* 165 (2023), 107327. doi:[10.1016/j.compbiomed.2023.107327](https://doi.org/10.1016/j.compbiomed.2023.107327)
- [40] Zhengzhen Li, Ke Li, and Na Wei. 2020. A sEMG-Based Hand Gesture Recognition Using Multichannel CNN and MLP. In *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 867–871. doi:[10.1109/CISP-BMEI51763.2020.9263646](https://doi.org/10.1109/CISP-BMEI51763.2020.9263646)
- [41] Xianle Lin, Xu Zhang, Xuan Zhang, Xiang Chen, and Xun Chen. 2023. DSDAN: Dual-step domain adaptation network based on bidirectional knowledge distillation for cross-user myoelectric pattern recognition. *IEEE Sensors Journal* 23, 21 (2023), 26765–26775. doi:[10.1109/JSEN.2023.3305619](https://doi.org/10.1109/JSEN.2023.3305619)
- [42] Miaomiao Liu, Sikai Yang, Wyssanie Chomsin, and Wan Du. 2022. Real-time tracking of smartwatch orientation and location by multitask learning. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. 120–133. doi:[10.1145/3560905.3568548](https://doi.org/10.1145/3560905.3568548)
- [43] Yilin Liu, Shijia Zhang, and Mahanth Gowda. 2021. NeuroPose: 3D hand pose tracking using EMG wearables. In *Proceedings of the Web Conference 2021*. 1471–1482. doi:[10.1145/3442381.3449890](https://doi.org/10.1145/3442381.3449890)
- [44] Yu Lu, Dian Ding, Ran Wang, and Guangtao Xue. 2024. HCMG: Human-Capacitance based Micro Gesture for VR/AR. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 766–770. doi:[10.1145/3675094.3678386](https://doi.org/10.1145/3675094.3678386)
- [45] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. 2020. Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27. doi:[10.1145/3380984](https://doi.org/10.1145/3380984)
- [46] Yongqiang Ma, Badong Chen, Pengju Ren, Nanning Zheng, Giacomo Indiveri, and Elisa Donati. 2020. EMG-based gestures classification using a mixed-signal neuromorphic processing system. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 10, 4 (2020), 578–587. doi:[10.1109/JETCAS.2020.3037951](https://doi.org/10.1109/JETCAS.2020.3037951)
- [47] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017. Echoflex: Hand gesture recognition using ultrasound imaging. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 1923–1934. doi:[10.1145/3025453.3025807](https://doi.org/10.1145/3025453.3025807)
- [48] Roberto Merletti and Dario Farina. 2016. *Surface electromyography: physiology, engineering, and applications*. John Wiley & Sons.
- [49] Xiang Mian, Zhou Bingtao, Cheng Shiqiang, and Liu Song. 2024. MCMP-Net: MLP combining max pooling network for sEMG gesture recognition. *Biomedical Signal Processing and Control* 90 (2024), 105846. doi:[10.1016/j.bspc.2023.105846](https://doi.org/10.1016/j.bspc.2023.105846)
- [50] Ali Moin, Andy Zhou, Abbas Rahimi, Simone Benatti, Alisha Menon, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, Fred Burghardt, et al. 2018. An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5. doi:[10.1109/ISCAS.2018.8351613](https://doi.org/10.1109/ISCAS.2018.8351613)
- [51] Mansooreh Montazerin, Farnoosh Naderkhani, and Arash Mohammadi. 2023. Spiking neural networks for semg-based hand gesture recognition. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 4859–4864. doi:[10.1109/SMC53992.2023.10393866](https://doi.org/10.1109/SMC53992.2023.10393866)
- [52] Gaudi Morantes, Gerardo Fernández, and Miguel Altuve. 2013. A threshold-based approach for muscle contraction detection from surface EMG signals. In *IX International Seminar on Medical Information Processing and Analysis*, Vol. 8922. SPIE, 90–100. doi:[10.1117/12.2035673](https://doi.org/10.1117/12.2035673)

- [53] Nadia Nasri, Sergio Orts-Escalano, Francisco Gomez-Donoso, and Miguel Cazorla. 2019. Inferring static hand poses from a low-cost non-intrusive sEMG sensor. *Sensors* 19, 2 (2019), 371. [doi:10.3390/s19020371](https://doi.org/10.3390/s19020371)
- [54] Javier Navallas, Mikel Ariz, Arantxa Villanueva, Javier San Agustin, and Rafael Cabeza. 2011. Optimizing interoperability between video-oculographic and electromyographic systems. *Journal of Rehabilitation Research & Development* 48, 3 (2011). [doi:10.1682/jrrd.2010.06.0112](https://doi.org/10.1682/jrrd.2010.06.0112)
- [55] Oymotion. [n. d.]. G-ForcePro+. <http://www.oymotion.com/product17/82>.
- [56] Long Peng, Zeng-Guang Hou, Nikola Kasabov, Gui-Bin Bian, Luige Vladareanu, and Hongnian Yu. 2015. Feasibility of NeuCube spiking neural network architecture for EMG pattern recognition. In *2015 International Conference on Advanced Mechatronic Systems (ICAMECHS)*. IEEE, 365–369. [doi:10.1109/ICAMECHS.2015.7287090](https://doi.org/10.1109/ICAMECHS.2015.7287090)
- [57] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 7784 (2019), 607–617. [doi:10.1038/s41586-019-1677-2](https://doi.org/10.1038/s41586-019-1677-2)
- [58] Sasha Salter, Richard Warren, Collin Schlager, Adrian Spurr, Shangchen Han, Rohin Bhasin, Yujun Cai, Peter Walkington, An- uoluwapo Bolarinwa, Robert Wang, Nathan Danielson, Josh Merel, Eftychios Pnevmatikakis, and Jesse Marshall. 2024. emg2pose: A Large and Diverse Benchmark for Surface Electromyographic Hand Pose Estimation. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 55703–55728. https://proceedings.neurips.cc/paper_files/paper/2024/file/64f8884f6ba3d9ace5bb647c4f917896-Paper-Datasets_and_Benchmarks_Track.pdf
- [59] Debajit Sarma and Manas Kamal Bhuyan. 2021. Methods, databases and recent advancement of vision-based hand gesture recognition for hci systems: A review. *SN Computer Science* 2, 6 (2021), 436. [doi:10.1007/s42979-021-00827-x](https://doi.org/10.1007/s42979-021-00827-x)
- [60] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I am a smartwatch and i can track my user’s arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*. 85–96. [doi:10.1145/2906388.2906407](https://doi.org/10.1145/2906388.2906407)
- [61] SICHIRAY. [n. d.]. EMGPRO. <https://sichiray-tech.yuque.com/dm0eyv/chapin/bflmvgtg46wszkm>.
- [62] G Staude and W Wolf. 1999. Objective motor response onset detection in surface myoelectric signals. *Medical engineering & physics* 21, 6-7 (1999), 449–467. [doi:10.1016/s1350-4533\(99\)00067-3](https://doi.org/10.1016/s1350-4533(99)00067-3)
- [63] Jesus Suarez and Robin R Murphy. 2012. Hand gesture recognition with depth images: A review. In *2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication*. IEEE, 411–417. [doi:10.1109/ROMAN.2012.6343787](https://doi.org/10.1109/ROMAN.2012.6343787)
- [64] Antong Sun, Xiang Chen, Mengjuan Xu, Xu Zhang, and Xun Chen. 2023. Feasibility study on the application of a spiking neural network in myoelectric control systems. *Frontiers in Neuroscience* 17 (2023), 1174760. [doi:10.3389/fnins.2023.1174760](https://doi.org/10.3389/fnins.2023.1174760)
- [65] Triwiyanto Triwiyanto, I Putu Alit Pawana, and Mauridhi Hery Purnomo. 2020. An improved performance of deep learning based on convolution neural network to classify the hand motion by evaluating hyper parameter. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28, 7 (2020), 1678–1688. [doi:10.1109/TNSRE.2020.2999505](https://doi.org/10.1109/TNSRE.2020.2999505)
- [66] Le Wu, Xu Zhang, Kun Wang, Xiang Chen, and Xun Chen. 2020. Improved high-density myoelectric pattern recognition control against electrode shift using data augmentation and dilated convolutional neural network. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28, 12 (2020), 2637–2646. [doi:10.1109/TNSRE.2020.3030931](https://doi.org/10.1109/TNSRE.2020.3030931)
- [67] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. 2018. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience* 12 (2018), 331. [doi:10.3389/fnins.2018.00331](https://doi.org/10.3389/fnins.2018.00331)
- [68] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 1311–1318. [doi:10.1609/aaai.v33i01.33011311](https://doi.org/10.1609/aaai.v33i01.33011311)
- [69] Chen Xie, Alessio Burrello, Francesco Daghero, Luca Benini, Andrea Calimera, Enrico Macii, Massimo Poncino, and Daniele Jahier Pagliari. 2023. Reducing the Energy Consumption of sEMG-Based Gesture Recognition at the Edge Using Transformers and Dynamic Inference. *Sensors* 23, 4 (2023), 2065. [doi:10.3390/s23042065](https://doi.org/10.3390/s23042065)
- [70] Kexin Xing, Peipei Yang, Jian Huang, Yongji Wang, and Quanmin Zhu. 2014. A real-time EMG pattern recognition method for virtual myoelectric hand control. *Neurocomputing* 136 (2014), 345–355. [doi:10.1016/j.neucom.2013.12.010](https://doi.org/10.1016/j.neucom.2013.12.010)
- [71] Yannan Xing, Gaetano Di Caterina, and John Soraghan. 2020. A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Frontiers in neuroscience* 14 (2020), 590164. [doi:10.3389/fnins.2020.590164](https://doi.org/10.3389/fnins.2020.590164)
- [72] Dezhen Xiong, Daohui Zhang, Xingang Zhao, and Yiwen Zhao. 2021. Deep learning for EMG-based human-machine interaction: A review. *IEEE/CAA Journal of Automatica Sinica* 8, 3 (2021), 512–533. [doi:10.1109/JAS.2021.1003865](https://doi.org/10.1109/JAS.2021.1003865)
- [73] Mengjuan Xu, Xiang Chen, Antong Sun, Xu Zhang, and Xun Chen. 2023. A novel event-driven spiking convolutional neural network for electromyography pattern recognition. *IEEE Transactions on Biomedical Engineering* 70, 9 (2023), 2604–2615. [doi:10.1109/TBME.2023.3258606](https://doi.org/10.1109/TBME.2023.3258606)
- [74] Xuhai Xu, Jun Gong, Carolina Brum, Lilian Liang, Bongsoo Suh, Shivam Kumar Gupta, Yash Agarwal, Laurence Lindsey, Runchang Kang, Behrooz Shahsavari, et al. 2022. Enabling hand gesture customization on wrist-worn devices. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19. [doi:10.1145/3491102.3501904](https://doi.org/10.1145/3491102.3501904)
- [75] Zengjie Yu, Yiqian Lu, Qi An, Cong Chen, Ye Li, and Yishan Wang. 2022. Real-time multiple gesture recognition: Application of a lightweight individualized 1D CNN model to an edge computing system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 30 (2022), 990–998. [doi:10.1109/TNSRE.2022.3165858](https://doi.org/10.1109/TNSRE.2022.3165858)

- [76] Marcello Zangheri, Simone Benatti, Alessio Burrello, Victor Kartsch, Francesco Conti, and Luca Benini. 2019. Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor. *IEEE transactions on biomedical circuits and systems* 14, 2 (2019), 244–256. doi:[10.1109/TBCAS.2019.2959160](https://doi.org/10.1109/TBCAS.2019.2959160)
- [77] Jilin Zhang, Mingxuan Liang, Jinsong Wei, Shaojun Wei, and Hong Chen. 2021. A 28nm configurable asynchronous SNN accelerator with energy-efficient learning. In *2021 27th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*. IEEE, 34–39. doi:[10.1109/ASYNC48570.2021.00013](https://doi.org/10.1109/ASYNC48570.2021.00013)
- [78] Yingwei Zhang, Yiqiang Chen, Hanchao Yu, Xiaodong Yang, Ruizhe Sun, and Bixiao Zeng. 2021. A feature adaptive learning method for high-density semg-based gesture recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–26. doi:[10.1145/3448114](https://doi.org/10.1145/3448114)
- [79] Maozheng Zhao, Alec M Pierce, Ran Tan, Ting Zhang, Tianyi Wang, Tanya R Jonker, Hrvoje Benko, and Aakar Gupta. 2023. Gaze speedup: Eye gaze assisted gesture typing in virtual reality. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 595–606. doi:[10.1145/3581641.3584072](https://doi.org/10.1145/3581641.3584072)
- [80] Tianming Zhao, Jian Liu, Yan Wang, Hongbo Liu, and Yingying Chen. 2019. Towards low-cost sign language gesture recognition leveraging wearables. *IEEE Transactions on Mobile Computing* 20, 4 (2019), 1685–1701. doi:[10.1109/TMC.2019.2962760](https://doi.org/10.1109/TMC.2019.2962760)
- [81] Chenlin Zhou, Liutao Yu, Zhaokun Zhou, Zhengyu Ma, Han Zhang, Huihui Zhou, and Yonghong Tian. 2023. Spikingformer: Spike-driven residual learning for transformer-based spiking neural network. *arXiv preprint arXiv:2304.11954* (2023). doi:[10.48550/arXiv.2304.11954](https://doi.org/10.48550/arXiv.2304.11954)
- [82] Muhammad Zia ur Rehman, Asim Waris, Syed Omer Gilani, Mads Jochumsen, Imran Khan Niazi, Mohsin Jamil, Dario Farina, and Ernest Nlandu Kamavuako. 2018. Multiday EMG-based classification of hand motions with deep learning techniques. *Sensors* 18, 8 (2018), 2497. doi:[10.3390/s18082497](https://doi.org/10.3390/s18082497)

A sEMG Acquisition System

A.1 Parameter of GBS Wristbands

At present, most sEMG wristbands have large electrodes and device volumes. We aspire to diminish the dimensions and weight of GBS wristbands, making them as convenient to wear as traditional watches or bands. Given the standard wristband width of 22 mm, the dimensions, spacing of electrodes, and circuit design are significantly constrained. Drawing on [19] and our experience, we have designed electrode surface areas of approximately 7 mm² and 20 mm², with a distance of 5–7 mm between each electrode pair and 12–20 mm between successive electrode pairs. This design ensures a high SNR and comprehensive coverage of wrist muscles. The GBS-I contained electrodes made of polymer flexible materials, with an area of contact of 20 mm² and a weight of 65 g. The band was made of silica gel. The GBS-II contained electrodes made of nanoscale silver paste, an area of contact of 7 mm² and a weight of 15 g. The band was made of flexible material. The sampling rate of the two wristbands was 2000 Hz and their bandwidth was 20–1,000 Hz, while they used BLE 5.1 for data transmission. A detailed comparison of the parameters of GBS wristbands with those of other sEMG acquisition devices on the market is shown in Table 8. In summary, compared to existing wristband devices, our design demonstrates significant improvements in signal quality and stability (e.g., higher sampling accuracy, higher sampling rate, and more stable Bluetooth transmission) as well as wearability (e.g., lighter weight). These advancements make our device more aligned with consumer-grade product standards and better suited for practical applications. Figure 17 illustrates the hardware structure schematics of the two wristbands used in this study. For further technical details, interested readers are encouraged to contact the authors directly.

A.2 Data Collection Process

The data collection process is shown in Figure 18. Each subject sat in front of the collection devices while wearing the GBS wristband and performed gestures as they were recorded by the three cameras. The PC first played a guiding video, following which the subject performed the corresponding gestures. The sEMG signals collected from the wristband were transmitted to the PC through Bluetooth. The PC then processed and displayed the sEMG signals of each channel in real-time. Three cameras recorded videos at different angles. When problems arose during the subsequent data analysis, the video could be used to track the subjects' actions. To ensure the accuracy of the ground truth labels for target action segments, the video recordings from the three cameras were

Table 8. Comparison of sEMG acquisition devices.

	Sichiray [61]	Oymotion [55]	Myo [37]	3DC [19]	GBS - I	GBS - II
sEMG channels	8	8	8	10	8	3
sEMG ADC	8 bits	8 bits	8 bits	10 bits	16 bits	16 bits
sEMG sampling rate	500 Hz	1000 Hz	200 Hz	1000 Hz	2000 Hz	2000 Hz
Bandwidth	20-500 Hz	20-500 Hz	5-100 Hz	20-500 Hz	20-1000 Hz	20-1000 Hz
Contact dimensions	150 mm ²	≈66 mm ²	100 mm ²	50 mm ²	20 mm ²	7 mm ²
electrodes material	Stainless steel	Stainless steel with silver coated	Stainless steel	Electroless nickelated immersion gold	Polymer materials	Silver
IMU sensors	Six-axis	Nine-axis	Nine-axis	Nine-axis	Six-axis	Nine-axis
IMU sampling rate	50 Hz	50 Hz	50 Hz	50 Hz	200 Hz	200 Hz
Transmitter	BLE 5.0	BLE 4.1	BLE4.0	Similar BLE	BLE 5.1	BLE 5.1
Weight	≈100 g	≈80 g	≈90 g	62 g	65 g	15 g

synchronized with the EMG signal timestamps. This synchronization allowed us to precisely extract and label the target action segments based on the subjects' actual movements. Additionally, the labeled segments were reviewed and validated by domain experts to further ensure their reliability. The labeled segments were then used as the ground truth for evaluating the performance of the TAD-LIF method and other comparison methods.

B Comparative Models

B.1 Details of Comparative Models

We compared the proposed DCSNN with numerous small-scale SOTA methods. The network architecture composition of each comparative model is detailed below.

CNN-1d [65]: Two CNN Blocks (a 1D convolutional layer, a batch normalization layer, and a ReLU function), a 1D max pooling layer, two CNN Block (a 1D convolutional layer, a batch normalization layer, and a ReLU function), a 1D average pooling layer, and two fully connected layers.

CNN-2d [82]: Two CNN Blocks (a 2D convolutional layer, a batch normalization layer, a ReLU function, a dropout function, and a 2D max pooling layer), and two fully connected layers.

GRU [53]: Three GRU Blocks (a GRU layer, a Tanh function, and a dropout function), and two fully connected layers.

SNN [64]: Three fully connected layers with LIF layers.

SCNN [73]: Two SCNN Blocks (a 2D convolutional layer, a 2D max pooling layer, and a LIF layer), and two fully connected layers with LIF layers.

MLP [40]: Three fully connected layers with ReLU function.

MCMP [49]: A fully connected layer with ReLU function, and a 1D max pooling layer.

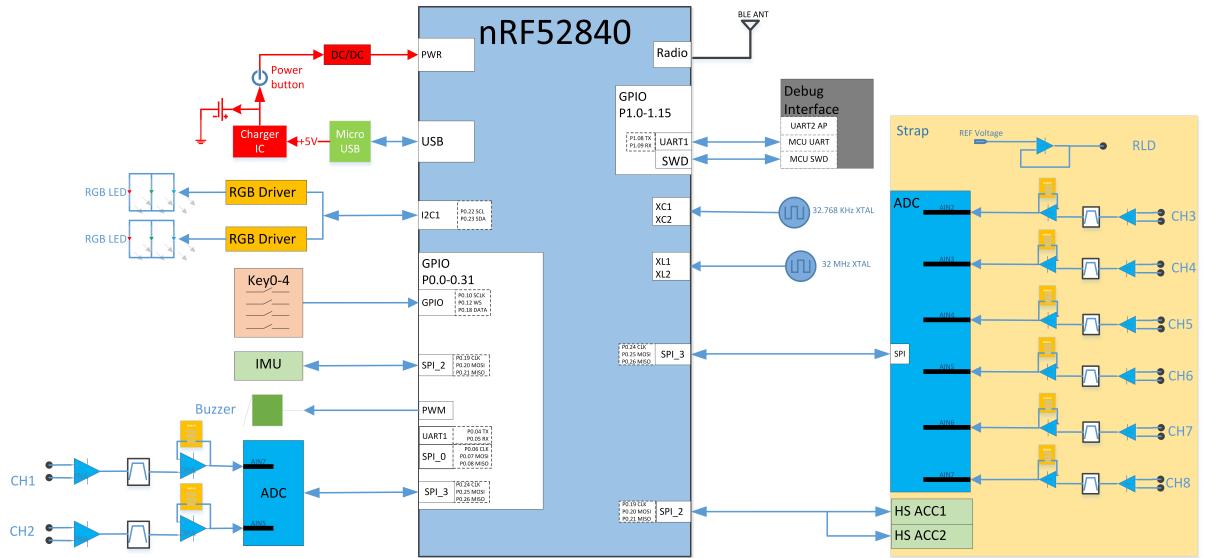


Fig. 17. Hardware schematics of GBS-I wristband.

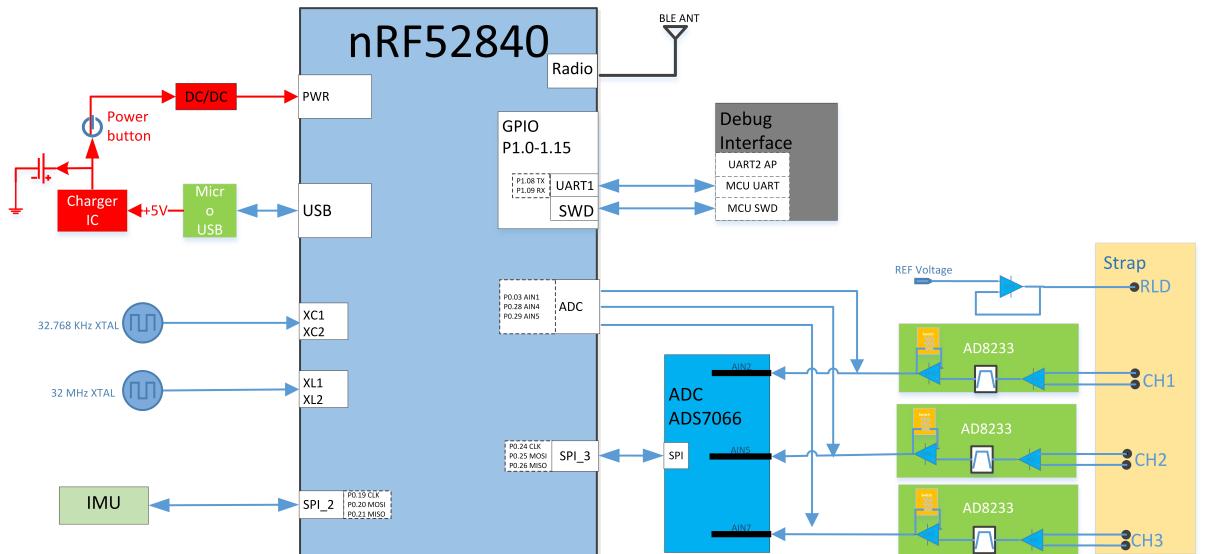


Fig. 18. Hardware schematics of GBS-II wristband.

B.2 Power Consumption of Models

When conducting inference in DL-based models, the main operations performed are accumulation (AC) and accumulation (MAC). We can estimate the power consumption of inferring one sample by calculating the required number of AC and MAC. Horowitz et al. showed that a 32-bit integer AC operation consumes about 0.1 pJ of

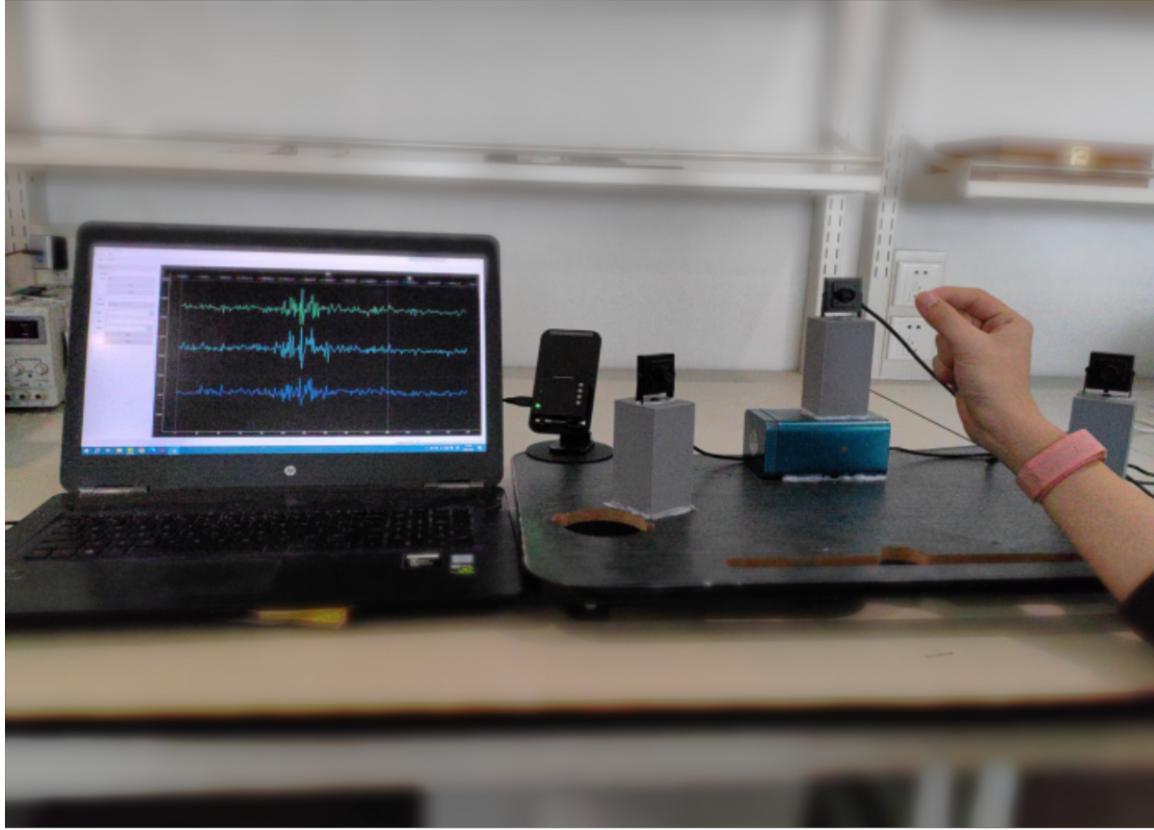


Fig. 19. Overview of devices used to collect sEMG signals. A PC was used to record the data, a dongle was used for data transmission, and three cameras were used to record the gestures at different angles.

power while the MAC operation consumes about 3.2 pJ on the 45 nm CMOS25 [29]. By using these metrics, we estimated the power consumed by different models when inferring one sample. The power consumption of one sample can be expressed as Eq.12. This method for calculating power consumption is referenced from [64, 73].

$$P = 3.1pJ * N_{MAC} + 0.1pJ * N_{AC} \quad (12)$$

We mainly calculated the power consumption of the convolutional layers, fully connected layers, GRU layers, LIF layers, and spike encoding for each method. Table 9 presents the calculation formulas for power consumption across each layer. The calculations in pooling layers, batch normalization layers, ReLU functions, and TAD-LIF can be ignored. T represents the signal length of a sample, W represents the number of sEMG electrodes, C represents the number of channels, k represents the kernel size, F represents the number of neurons in the fully connected layer, H represents the hidden size in the GRU layer, N represents the number of spike trains.

Received 1 November 2024; revised 5 February 2025; accepted 5 April 2025

Table 9. The calculation formulas for inference power.

Layers & Functions	N_{AC}	N_{MAC}
Convolutional layer (input float trains)	$T * W * C_{out} * (k_1 * k_2 * C_{in} - 1)$	$k_1 * k_2 * T * W * C_{in} * C_{out}$
Fully connected layer (input float trains)	$F_{out} * (F_{in} - 1)$	$F_{in} * F_{out}$
Convolutional layer (input spike trains)	$T * W * C_{out} * (k_1 * k_2 * C_{in} * 2 - 1)$	/
Fully connected layer (input spike trains)	$F_{out} * (F_{in} * 2 - 1)$	/
GRU layer	$7 * H$	$H * (3 * T * W + 8 * H)$
LIF layer	$2 * T * W$	$3 * T * W$
Spike encoding	$N * T$	/