Iterative Self-Training Based Domain Adaptation for Cross-User sEMG Gesture Recognition

Kang Wang, Yiqiang Chen, *Senior Member, IEEE*, Yingwei Zhang, Xiaodong Yang, and Chunyu Hu

***Abstract*— Surface electromyography (sEMG) based gesture recognition has received broad attention and application in rehabilitation areas for its direct and fine-grained sensing ability. sEMG signals exhibit strong user dependence properties among users with different physiology, causing the inapplicability of the recognition model on new users. Domain adaptation is the most representative method to reduce the user gap with feature decoupling to acquire motion-related features. However, the existing domain adaptation method shows awful decoupling results when handling complex time-series physiological signals. Therefore, this paper proposes an Iterative Self-Training based Domain Adaptation method (STDA) to supervise the feature decoupling process with the pseudo-label generated by self-training and to explore cross-user sEMG gesture recognition. STDA mainly consists of two parts, discrepancy-based domain adaptation (DDA) and pseudo-label iterative update (PIU). DDA aligns existing users' data and new users' unlabeled data with a Gaussian kernel-based distance constraint. PIU Iteratively continuously updates pseudo-labels to generate more accurate labelled data on new users with category balance. Detailed experiments are performed on publicly available benchmark datasets, including the NinaPro dataset (DB-1 and DB-5) and the CapgMyo dataset (DB-a, DB-b, and DB-c). Experimental results show that the proposed method achieves significant performance improvement compared with existing sEMG gesture recognition and domain adaption methods.**

Manuscript received 10 December 2022; revised 1 July 2023; accepted 3 July 2023. Date of publication 7 July 2023; date of current version 26 July 2023. This work was supported in part by the National Key Research and Development Plan of China under Grant 2021YFC2501202, in part by the National Natural Science Foundation of China under Grant 61972383 and Grant 62202455, in part by the Beijing Municipal Science & Technology Commission under Grant Z221100002722009, and in part by the Innovative Research Program of Shandong Academy of Intelligent Computing Technology under Grant SDAICT2191010. *(Corresponding author: Yiqiang Chen.)*

Kang Wang is with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China, and also with the Shangdong Academy of Intelligent Computing Technology, Jinan 100190, China (e-mail: 10431200629@stu.qlu.edu.cn).

Yiqiang Chen, Yingwei Zhang, and Xiaodong Yang are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100045, China, also with the Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing 100190, China, also with the University of Chinese Academy of Sciences, Beijing 101408, China, and also with the Shangdong Academy of Intelligent Computing Technology, Jinan 100190, China (e-mail: zhangyingwei@ict.ac.cn; yqchen@ict.ac.cn; yangxiaodong@ict.ac.cn).

Chunyu Hu is with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China (e-mail: hcy@qlu.edu.cn).

Digital Object Identifier 10.1109/TNSRE.2023.3293334

***Index Terms*— Surface electromyography, gesture recognition, cross-user, domain adaptation, semi-supervised learning.**

I. Introduction

AS THE most flexible part of the human body, the hand can complete a variety of complex movements and is essential to an individual's life, study and work [36]. Rehabilitation of hand function after an injury or chronic disease is crucial in rehabilitation medicine because it affects not only hand movement but also other related body parts (i.e., wrist, upper limb) and even cognitive abilities (i.e., working memory, executive function, and thinking ability) [19], [38]. Among various hand rehabilitation techniques, surface electromyography (sEMG) based gesture recognition is a crucial technology to measure human behavioural ability and further perceive human intentions [16]. Furthermore, this technology has been widely studied because it can directly perceive and analyze human upper limb muscle activity [48]. Gesture recognition based on sEMG has a wide range of applications. For example, intelligent prostheses based on sEMG can help upper limb amputees regain some of their original function [35] and promote cognitive improvement.

sEMG gesture recognition is a multi-classification machine learning problem. As early as 2008, Kim et al. [20] proposed using traditional machine learning methods for sEMG gesture recognition, with an accuracy rate of 94%. Tsinganos et al. [37] pointed out that with the robust feature learning ability of deep learning, many researchers have also achieved high recognition accuracy through models such as convolutional neural networks in recent years. Although high accuracy rates have been reported in laboratory settings, most have not been translated into clinical applications [49]. The main reason for the failure is the inconsistency of statistics. Several factors contribute to the inconsistent distribution of data. The main reasons are limb position offset, electrode offset, and especially physiological differences between users (such as muscle fatigue, skin resistance, and muscle strength). The above factors lead to the failure of the previously trained model to adapt to new users. Even if the same movement is performed, the signals from different users will significantly differ.

Domain adaptation is an effective method to solve the inconsistency between two data distributions [11]. Domain adaptation is derived initially from computer vision and has achieved splendid results. The success of domain adaptation in computer vision may be due to the potential label correlation between the source domain and target domain images. However, sEMG signals from two different people with the same movement did not show a significant correlation. The existing domain adaptation methods have poor adaptation effects on sEMG signals and fail to decouple the information related to motion well. Only a limited number of gestures can be recognized, their validity has yet to be verified on public datasets [49], or a certain amount of target domain data is required for adaptation [31].

Therefore, this paper considers adding a small amount of supervision information to guide the decoupling process and proposes an Iterative Self-Training based Domain Adaptation to explore cross-user sEMG gesture recognition, namely STDA. In detail, STDA includes two parts: discrepancy-based domain adaptation (DDA) and pseudo-label iterative update (PIU). DDA uses a Gaussian kernel-based distance constraint to reduce the distance between source and target domains. PIU uses an iterative self-training loop to produce more accurate pseudo labels. Overall, the STDA method outperforms the state-of-the-art on the five sub-datasets. Our method provides over 25% improvement over baseline, over 5% over supervised domain adaptation, and over 24% over unsupervised domain adaptation.

The significant advantages of the proposed method are as follows: 1) We propose a novel data decoupling method, which uses self-training to supervise domain adaptation and achieves marvellous decoupling results. 2) We propose an adjustment mechanism to deal with the pseudo-label imbalance during model training, which makes the model more accurate. 3) Experimental results on five publicly available datasets show that our model generally outperforms state-of-the-art methods for sEMG gesture recognition and domain adaptation.

The remainder of this paper is organized as follows. Section II overviews existing sEMG gesture recognition, self-training, and transfer learning methods. Section III gives the formulation of the problem. Section IV sets out our motivation. Section V presents the STDA method and architecture diagram. Section VI gives the experimental setup and analysis. Finally, Section VII summarizes the entire paper and gives plans.

II. Related Work

The proposed STDA method is mainly related to sEMG gesture recognition, self-training, and transfer learning. Therefore, this section will review the latest research in these fields and their intersection.

A. sEMG Based Gesture Recognition

According to the sensing solution, gesture recognition can be divided into vision-based, accelerometer-based, data glove-based, sEMG-based, WiFi-based, radar-based, etc [2], [3], [14], [32], [33], [40]. Simultaneously, according to the performing procedure, the gesture contains two types, i.e., the static gesture and the dynamic gesture. For example, three subsets of CapgMyo (including DB-a, DB-b and DB-c) are static gestures, and two subsets of NinaPro (including DB-1 and DB-5) are dynamic gestures. Compared with other gesture recognition solutions, the one based on sEMG has many advantages. sEMG signal can perceive and analyze human muscle activity directly, and sEMG-based gesture recognition is also insensitive to light. Thus, sEMG gesture recognition has wide applications, especially in the rehabilitation medicine area. For example, Wang et al. [44] used sEMG signals to capture human motion intentions, converted them into control commands and sent them to the prosthetic control system to complete the grasp of objects, with an average accuracy of about 96%. Cruz-Sá nchez et al. [9] used the signals collected by the myoelectric armband as the command to control the hand exoskeleton to assist in recovering hand impairment. Experimental results showed that an accuracy of about 81% was obtained using the K-nearest neighbour classification algorithm. Oñ a et al. [26] explored the practicability of using sEMG signals to control games to help young people with multiple sclerosis. Adebayo et al. [1] used the recurrent neural network (RNN), long short-term memory (LSTM), based on sEMG signals to control electric wheelchairs to help disabled and older people become unaided in life.

Generally speaking, we can model sEMG gesture recognition as a machine learning process, and the processing flow includes pre-processing, feature extraction, and classification. As sEMG signals have the non-stationary property and their intrinsic characteristics can not be expressed well with a separate domain, some time-frequency transformation methods (e.g., Fourier transforms and Wavelet transforms) are generally used to convert the original signals into time-frequency maps [27], [28]. Besides the traditional machine learning methods that extract sEMG features and construct classification models separately, many end-to-end deep learning models have been applied to sEMG gesture recognition, including the convolutional neural network (CNN) and recurrent neural network (RNN). At the same time, some of their variants and improvements have also been created. Wang et al. [40] proposed a variant method of CNN to improve gesture recognition accuracy by optimizing the convolution kernel. The experiments in Ninapro DB-1 showed recognition improvement compared with the traditional CNN method. Bai et al. [5] fused the CNN and LSTM methods to build a convolutional recurrent neural network that achieved an accuracy of about 91%. Recently, attention-based models have also been employed for sEMG gesture recognition. Josephs et al. [15] proposed a model based on an attention mechanism that can outperform previous complex models based on CNN in the recognition task with 53 gestures. Lv et al. [24] proposed a deep learning architecture that fused the attention mechanism and CNN and achieved an accuracy higher than 97% on their own and public-available datasets.

However, influenced by the distribution differences of different users' sEMG signals, the sEMG gesture recognition model encounters severe user-dependent problems. Specifically, if the model established by existing users is directly applied to new users, the accuracy will drop significantly. Zhang et al. [50] reported only about 40% recognition accuracy on cross-user gesture recognition tasks involving five gestures. Similarly, Kanoga et al. [17] reported a similar low result on cross-user recognition tasks. Ketykó et al. [18] reported that gesture recognition accuracy drops below 50% in the presence of domain shift.

B. Self-Training

In recent years, due to the available large amount of labelled data, supervised learning has been successfully applied in many fields, such as computer vision, natural language processing, human activity recognition, sEMG gesture recognition, etc. However, acquiring large amounts of labelled data in real-world scenarios is labour-intensive and material-intensive, especially in medical systems.

The emergence of semi-supervision is to solve the above difficulties [52]. Semi-supervised learning mainly thinks about constructing a model with a small amount of supervised data while using other unlabeled data. Although there are no labels for unlabeled data, a large amount of data can provide additional distributional information, which is still beneficial for model training. Generally speaking, semi-supervised learning can be roughly divided into the following four categories: self-training-based, entropy regularization-based, clustering-based, and graph-based.

Semi-supervised learning relies on model assumptions, and when the model assumptions are correct, unlabeled samples can help improve learning performance. The commonly-used model assumptions are as follows:

- **Smoothness Assumption** [25]: The class labels of two closely spaced examples in a dense data region are similar. When two examples are connected by an edge in a lush data region, they have a high probability of having the same class label.
- **Clustering Assumption** [21]: When two samples are in the same cluster, they have the same class label with a high probability.
- **Manifold Assumption** [39]: If embedding high-dimensional data into a low-dimensional manifold, two examples have similar class labels when they lie within a small local neighbourhood in the low-dimensional manifold.

Self-training is a simple and effective semi-supervised learning method. The method based on self-training is simple and effective and is widely used. For example, Sahito et al. [34] used self-training to solve the image classification problem and evaluated it on three benchmark datasets. Yang et al. [47] proposed a teacher-student model to generate image captions. Bi et al. [6] used self-training to realize human activity recognition, advancing research in digital health. The general process of self-training can be summarized as follows: First, train a model with labelled data; Second, use the trained model to predict unlabeled data, usually using the head of the model (i.e., a linear classifier), and the resulting label is called a pseudo-label; Then, select reliable labelled data with a specific strategy, such as selecting samples with higher confidence; Finally, the model is retrained on the labelled data along with the pseudo-labelled data, and so on until the model converges.

Some researchers have also explored sEMG gesture recognition using semi-supervised learning. Guo et al. [13] achieved about 93% accuracy on single-finger movements using a semi-supervised learning algorithm. Xu et al. [46] explore using semi-supervised learning to control hand orthoses to help stroke patients. Du et al. [10] achieved decent performance on three public datasets using semi-supervised learning.

C. Transfer Learning

Besides the dependence on a large amount of labelled data, the success of traditional machine learning methods is also based on the assumption that the training and the testing data obey independent and identically distributed. To meet the challenges brought by the time and labour consumption of data labelling, transfer learning is proposed and increases wide attention. Transfer learning defines a new machine learning paradigm, which can learn knowledge from the source tasks and transfer the learned knowledge to the target tasks [29]. Recently, the more challenging problem of domain generalization has attracted the interest of researchers, where one or more distinct but related domains are known, and the goal is to generalize directly to an unseen domain [41].

Specifically, transfer learning refers to a given source domain and source task. The purpose is to use the source domain and source task to help it learn a good model on the target domain, satisfying that the source domain is not equal to the target domain or the source task is not equal to the target task [29]. Transfer learning can be divided into three categories: instance-based transfer, parameter sharing-based transfer, and feature-based transfer. Instance-based transfer focuses on selecting the most favourable samples from the source domain and re-weighting them. Parameter-based transfer focuses on how to find standard parameters and prior distributions between the two. Feature-based transfer focuses on how to find the shared feature space of the source and target domains. The feature-based transfer has received more attention and can be further refined into the following sub-categories: discrepancy-based, adversarial-based, and reconstruction-based [42]. Discrepancy-based transfer learning aligns the source and target domains in feature space; Adversarial-based transfer learning utilizes the idea of the Generative Adversarial Network to construct a feature extractor and a domain discriminator for adversarial training; Reconstruction-based transfer learning constructs a reconstruction task based on the auto-encoder to ensure that the features are invariant.

Transfer learning has been successfully applied in many fields, such as natural language processing, computer vision, medical health, and so on. For example, Prottasha et al. [30] used BERT-based transfer learning for sentiment analysis. Ghorbanali et al. [12] explored sentiment analysis based on CNN-based transfer learning. Ayana et al. [4] used transfer learning to classify breast cancer images. Li et al. [22] explored the use of transfer learning for pest image classification. In this paper, we focus on the research of transfer learning on sEMG gesture recognition. Many factors, such as electrode shift and user differences, can lead to inconsistencies in data distribution. Campbell et al. [7] proposed a neural network architecture based on batch normalization and adversarial transfer learning strategies. With a small amount of target user data provided, the classification accuracy of healthy users is greater than 86%, and the classification accuracy of disabled users is greater than 64% on ten gestures. Zhang et al. [49] proposed a feature-alignment transfer learning method based on an adaptive sampling method, which achieved an average accuracy of about 90% on the six gestures collected. Rahimian et al. [31] also reported that experiments on Ninapro DB-5 showed that in cross-session scenarios, the accuracy of domain adaptation decreases as the number of samples given by the target domain decreases. Zheng et al. [51] also reported that the model for existing users could not be directly generalized to new users, and they proposed an adaptive K-nearest neighbour algorithm that achieved about 68%, 73%, and 83% classification accuracy on 12, 8, and 4 gestures, respectively. The 2-step domain adaptation proposed by Ketykó et al. [18] achieves a classification accuracy of about 65% on the NinaPro DB-1 dataset. Chan et al. [8] reported that electrode shift decreased recognition accuracy and proposed an unsupervised domain adaptation method to achieve an accuracy improvement of about 8%.

However, in cross-session and cross-user scenarios, it is still challenging to build a model with good generalization performance and many gestures without using target annotation data or with little target annotation data.

III. Problem Formulation

The proposed STDA method aims to explore the effective usages of semi-supervised domain adaptation on sEMG gesture recognition problem. The data from all existing users constitute the source domain $\mathcal{S}$, where $\mathcal{S} = \left\{(\boldsymbol{x}_i^s, y_i^s)\right\}_{i=1}^{N_s}$, N_s represents the number of samples in the source domain, $\boldsymbol{x}_i^s$ indicates the ith sample of the source domain, and y_i^s indicates the label corresponding to the ith sample of the source domain. The data from the target user constitutes the target domain $\mathcal{T}$, $\mathcal{T} = \mathcal{T}_l \bigcup \mathcal{T}_u \bigcup \mathcal{T}_{te}$. $\mathcal{T}_l$ represents a small amount of labelled data in the target domain (for a C ($C \in \mathbb{N}^+$) classification problem, it has only C samples), $\mathcal{T}_u$ represents a relatively large amount of unlabeled data in the target domain, and $\mathcal{T}_{te}$ represents the target domain test data. The three of them satisfy the identically independent distribution. And the number of samples in $\mathcal{T}_u$ is larger than that in $\mathcal{T}_l$, i.e., $|\mathcal{T}_u| > |\mathcal{T}_l|$. $\mathcal{T}_l = \left\{(\boldsymbol{x}_j^{tl}, y_j^{tl})\right\}_{j=1}^{N_l}$, where N_l represents the number of labelled samples in the target domain. $\mathcal{T}_u = \left\{(\boldsymbol{x}_k^{tu})\right\}_{k=1}^{N_u}$, where N_u represents the number of unlabelled samples in the target domain. $\mathcal{T}_{te} = \left\{(\boldsymbol{x}_m^{te})\right\}_{m=1}^{N_{te}}$, where N_{te} represents the number of test samples in the target domain. In cross-user scenarios, there is a domain shift, i.e., the distribution is inconsistent ($P_{\mathcal{S}} \neq P_{\mathcal{T}}$). The above $\boldsymbol{x}_s, \boldsymbol{x}_t$ are the data vector, i.e., $\boldsymbol{x}_s, \boldsymbol{x}_t \in \mathbb{R}^{(d_1 \times d_2 \times d_3 \times d_4)}$. The above y_s, y_t are the category label, i.e., $y_s, y_t \in \mathbb{N}$. The goal of cross-user sEMG gesture recognition is to use the data of existing users in the source domain $\mathcal{S}$, a small amount of labelled data in the target domain $\mathcal{T}_l$, and relatively large amount of unlabeled data in the target domain $\mathcal{T}_u$ to obtain accurate labels, $\{y_m\}_{m=1}^{N_{te}}$, on the test data $\mathcal{T}_{te}$.

IV. Motivation

A. User Dependent Properties of sEMG Signal

Compared with the distribution differences among different domains in the standard machine learning settings, the sEMG signal shows more serious user-dependent properties among users (i.e., domains). As shown in Fig. 1(a), we use t-distributed stochastic neighbourhood embedding (t-SNE) to reduce two users' high-dimensional sEMG data features extracted by deep neural networks to two dimensions for visualization. The square represents the features of the first user, and the circle represents the features of the second user. The same colour represents the same gesture. It can be seen from the observation that under the same gesture, the features between different users are far apart. It shows that sEMG data has strong user dependence.

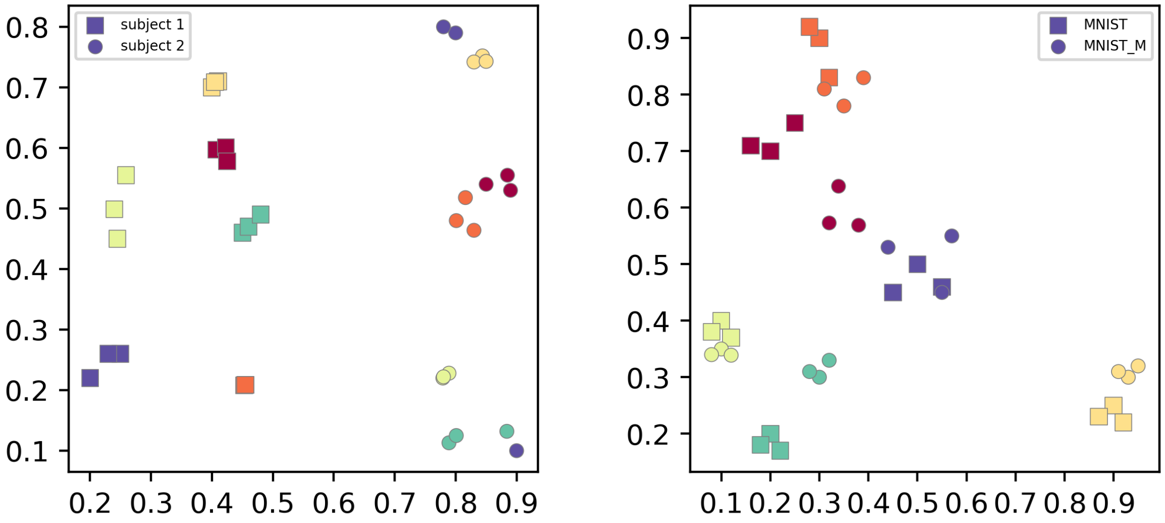
In addition, we also visualize the distribution differences in the standard machine learning settings. As shown in Fig. 1(b), the data of MNIST and MNIST_M are also reduced to two dimensions for visualization. The square represents the MNIST dataset, the circle represents the MNIST_M dataset, and the same colour represents the consistent category. As can be seen from the graph, the same categories are close to each other. From the analysis above, we can see that sEMG signals suffer more user-dependent severe properties.

B. Pre-Experiment With Unsupervised Domain Adaptation

Unsupervised domain adaptation is an effective method to solve domain shift, which has been widely studied and made good progress in computer vision and natural language processing. We have briefly explored this on the sEMG dataset, and we know that sEMG signals are mainly composed of two parts, the user-related and motion-related parts. We attempt to address the cross-user problem through unsupervised domain adaptation. We map the source and target domain signals into a shared space, hoping to train a user-independent classifier only related to gesture actions. However, the results could be better. We also tried an adversarial-based approach, and the results could have been better. We also tried an adversarial-based approach, and the results were not ideal either. We present partial experimental results of feature alignment based on kernel space distance, as shown in Fig 3. Observing the graph, we can see that the Maximum Mean Discrepancy (MMD) distance is regarded as the loss of model optimization, and it does not decrease but increases on the sEMG dataset. It shows that sEMG data has very extreme user dependence. Based on this, we believe the user's personalized information is still relatively critical. We still introduce very little labelled data of the target domain and have completed the personalized migration of target users. It should be emphasized that we use self-training to reduce the training burden of new users on the one hand. On the other hand, it is more conducive to domain adaptation to decouple action-related information.

V. Iterative Self-Training Based Domain Adaptation Method

Motivated by the pre-experiment above, we propose the Iterative Self-Training-based Domain Adaptation method (STDA).



(a) sEMG signal

(b) Image data

Fig. 1. T-SNE embedding of (a) sEMG signal and (b) Image data. The shape represents the domain, i.e., squares represent data from the first domain, and circles represent data from the second domain. The colour represents the class, i.e., the different gestures or different types of pictures.

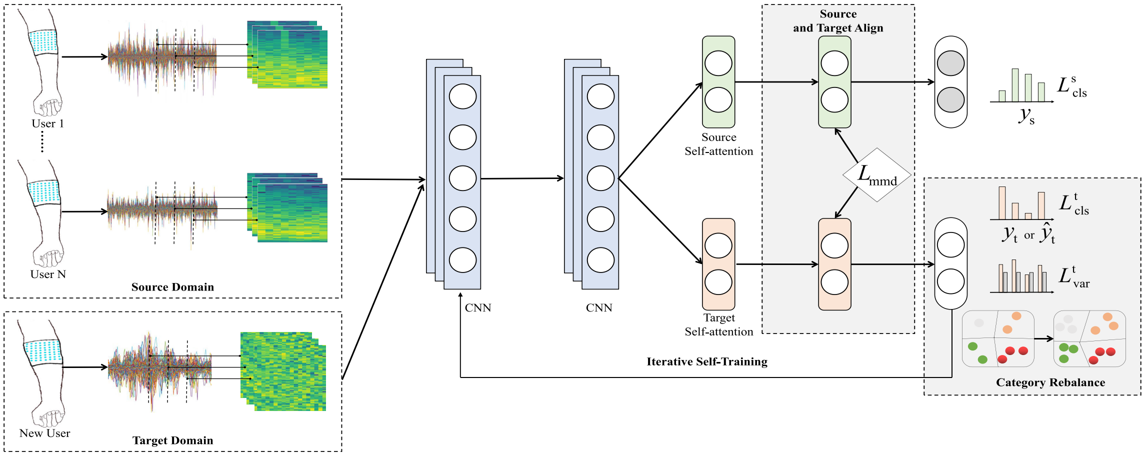
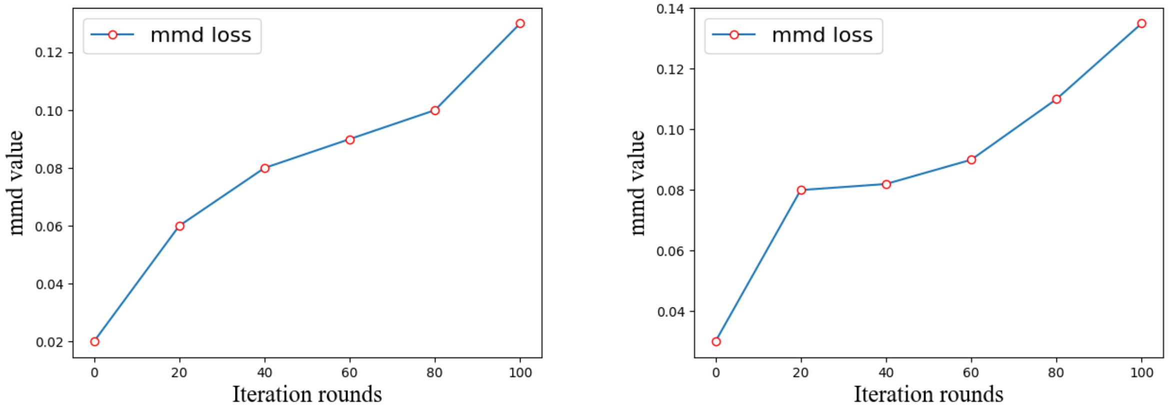


Fig. 2. Framework of the proposed STDA. STDA mainly consists of two parts, discrepancy-based domain adaptation (DDA) and pseudo-label iterative update (PIU). DDA aligns source and target domain features via distance constraints, and PIU iteratively generates target domain pseudo-labels.



(a) CapyMyo DB-a

(b) CapyMyo DB-c

Fig. 3. Trend of mmd loss during training.

This section will present the proposed STDA method. The framework of STDA is shown in Fig. 2. The method generally has two parts, discrepancy-based domain adaptation (DDA) and pseudo-label iterative update (PIU). DDA pursues the feature alignment of source and target domains, and PIU pursues iterative self-training. It should be emphasized that our model did not use the data used in the training phase during the test phase. In the following part, we will first give the feature alignment on the source and target domains in Section V-A. Second, we introduce iterative self-training in Section V-B. Then, we propose category rebalancing after each iteration in Section V-C. Finally, we give the implementation process of the algorithm in Section V-D.

A. Source and Target Align

Although the distribution of the source domain and the target domain is inconsistent in the original data space (as shown in Fig. 4), it is possible to narrow the distance between the source domain and the target domain in specific feature spaces. The goal is to realize that the model trained in the source domain performs excellently in the target domain. Discrepancy-based domain adaptation is a simple and effective method widely adopted in text data, image data, time series signals, etc. Therefore, in a cross-user scenario, the existing user's labelled data and the new user's unlabeled data can be aligned to realize gesture recognition.

The source domain can be denoted as $\mathcal{S} = \left\{(\boldsymbol{x}_i^s, y_i^s)\right\}_{i=1}^{N_s}$. The target domain can be denoted as $\mathcal{T} = \left\{(\boldsymbol{x}_j^t)\right\}_{j=1}^{N_t}$. In addition, the input space (i.e. X) and the label space (i.e. Y) are consistent, but the probability distributions of the two are inconsistent, i.e., $P_s \neq P_t$. The purpose of feature alignment is to learn a good mapping (f) to simultaneously map the source and target domains to a shared space ($\mathcal{H}$), in which the distance between the two is relatively close ($||D_s - D_t|| < \epsilon$ ($\epsilon \in \mathbb{R}, \epsilon > 0$)). D_s represents the distance from the source domain sample point to the origin in high-dimensional space, and D_t represents the distance from the target domain sample point to the origin.

The kernel learning method based on maximum mean discrepancy effectively measures the distribution difference. Maximum mean discrepancy measures the distance between two distributions in Reproducing Kernel Hilbert Space (RKHS). The distance between two distributions, P_s and P_t, can be defined as follows:

$$D(\mathcal{F}, P_s, P_t) = sup_{f\in\mathcal{F}}(E_{\boldsymbol{x}_s\sim P_s}[f(\boldsymbol{x}_s)] - E_{\boldsymbol{x}_t\sim P_t}[f(\boldsymbol{x}_t)]) \tag{1}$$

where $\mathcal{F}$ represents a function set under RKHS, $E_{\boldsymbol{x}\sim\cdot}$ represents the expectation under the source or target domain, $\boldsymbol{x}_s$ represents a sample point in the source domain, and $\boldsymbol{x}_t$ represents a sample point in the target domain. When the distribution of the source and target domains is close, the distance D approaches 0. The MMD between the source and target domains can be calculated as:

$$\begin{aligned} MMD(\mathcal{S}, \mathcal{T}) &= \left\| \frac{1}{N_s}\sum_{i=1}^{N_s}\phi(\boldsymbol{x}_i^s) - \frac{1}{N_t}\sum_{j=1}^{N_t}\phi(\boldsymbol{x}_j^t) \right\|_{\mathcal{H}} \\ &= \left(\frac{1}{{N_s}^2}\sum_{i=1}^{N_s}\sum_{i'=1}^{N_s} k(\boldsymbol{x}_i^s, \boldsymbol{x}_{i'}^s) - \frac{2}{N_s N_t} \right. \\ &\quad \left. \sum_{i=1}^{N_s}\sum_{j=1}^{N_t} k(\boldsymbol{x}_i^s, \boldsymbol{x}_j^t) + \frac{1}{{N_t}^2}\sum_{j=1}^{N_t}\sum_{j'=1}^{N_t} k(\boldsymbol{x}_j^t, \boldsymbol{x}_{j'}^t) \right)^{\frac{1}{2}} \end{aligned} \tag{2}$$

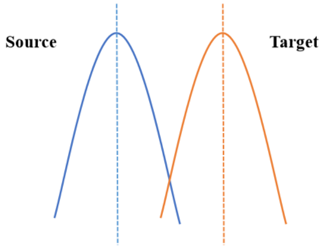


Fig. 4. Domain shift on source and target domains.

where ϕ represents a function that maps the original data to $\mathcal{H}$, $\phi(\boldsymbol{x}) = k(\cdot, \boldsymbol{x})$, and k generally takes the Gaussian kernel function. The distribution difference between two domains in a batch of samples can be measured with the distance in this high-dimensional space.

We know that the shallow layers of deep learning can learn general features, while the features learned by high layers are task-specific. Therefore, the MMD distance is often regarded as a loss, embedded in a particular layer of the inverse of the deep network and then optimized. The above is the alignment of the source domain and the target domain.

B. Iterative Self Training

Labelling data is time-consuming and labour-intensive and will seriously increase users' burdens. In recent years, self-training has been successfully applied in many fields to alleviate the dilemma of insufficient labelled data. Self-training has received constant attention as an effective semi-supervised learning method. The basic idea is to train a classifier with a small amount of labelled data. The classifier predicts the unlabeled data, and the prediction result is called a pseudo-label (as shown in Fig. 5), and then the two are combined to train the model. Specifically, the self-training process in this paper is as follows:

1) Train a model with a small amount of labelled data;
2) Use the trained model to predict the class labels of unlabeled samples;
3) Use a threshold to select pseudo-labels whose confidence level satisfies the condition;
4) Train the model jointly with labelled and pseudo-labelled data, and repeat 1)-4) until the model converges.

The training of the source domain model is a multi-classification problem. For multi-classification problems, cross-entropy is often used, and the calculation formula is as follows:

$$L_{log}(Y, \hat{Y}) = -\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{k=1}^{C} y_{i,k} log p_{i,k} \tag{3}$$

where Y represents the actual label, $\hat{Y}$ represents the predicted output, C represents the total number of classifications, and $p_{i,k}$ represents the probability that the ith sample is predicted to be the kth class. For each sample in a multi-class (C classification) problem, $\boldsymbol{x}_k^{tu} \in \mathcal{T}_u$, and the pseudo-labels (based on softmax confidence) are calculated as follows:

$$\hat{y} = \begin{cases} c(c \in C), \; if \; thres(\boldsymbol{x}_k^{tu}) > threshold; \\ \quad pass, \; else. \end{cases} \tag{4}$$

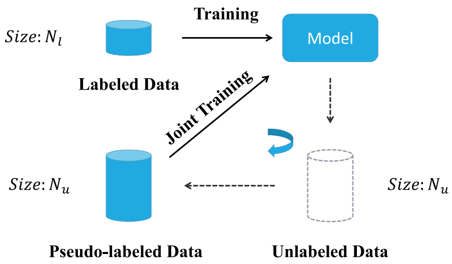


Fig. 5. Self-training process. The amount of unlabeled data is greater than the amount of labeled data, that is, $N_u > N_l$. Each iteration of unlabeled data re-labels all the data.

The algorithm flow of self-training is presented in Algorithm 1. The self-training proposed in this paper uses very few labeled data to train the model, labels the unlabeled data with the model, and then conducts joint training until the model converges.

Algorithm 1 Self-Training Process

```
Input: $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, $\mathcal{T}_u = \{(x_k^{tu})\}_{k=1}^{N_u}$
Output: $f$
repeat
    $f \leftarrow \mathcal{S}$    // Train the model
    $U' \leftarrow \{f(x), x \in \mathcal{T}_u\}$    // Predict unlabeled data
    $U_{pse} \leftarrow \{thres(x), x \in U'\}$    // Generate pseudo labels
    $\mathcal{S} \leftarrow \mathcal{S} \cup U_{pse}$    // Combine the two data
until convergence $\leftarrow f$
```

C. Category Re-Balance

If the categories are unbalanced in image recognition, we know the model will be adversely affected. During the experiment, we found that such a problem exists in the surface EMG data set. Fig. 6 shows a histogram of unprocessed pseudo-label categories after a certain iteration in the training process, showing extreme imbalance across categories. If this adequately deals with this effect, the model's accuracy can be improved. A pseudo-labelled class with many samples is called a majority class, and a pseudo-labelled class with a small number of samples is called a minority class. This paper adopts an oversampling method to make the minority class samples comparable to the majority class samples through oversampling.

At the same time, to keep the original public dataset from losing its balance property during training, we define a balance loss, which is equivalent to a variance. Assuming that the number of output gesture categories in a set is $x_1, x_2, \cdots, x_n$, the equilibrium loss is calculated as follows:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n}$$
$$var = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n} \tag{5}$$

Optimizing var as part of the overall optimization loss minimizes it to prevent violating the balance properties of the public dataset itself.

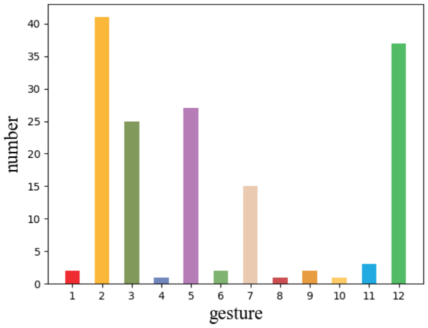


Fig. 6. Category imbalance on 12 gestures.

D. Method Implementation

The whole process of the STDA method proposed in this paper is described in Algorithm 2, which is a cross-user framework based on sEMG signals. The framework mainly involves three types of losses, including multi-class loss, maximum mean discrepancy loss, and equilibrium loss.

VI. Experiment

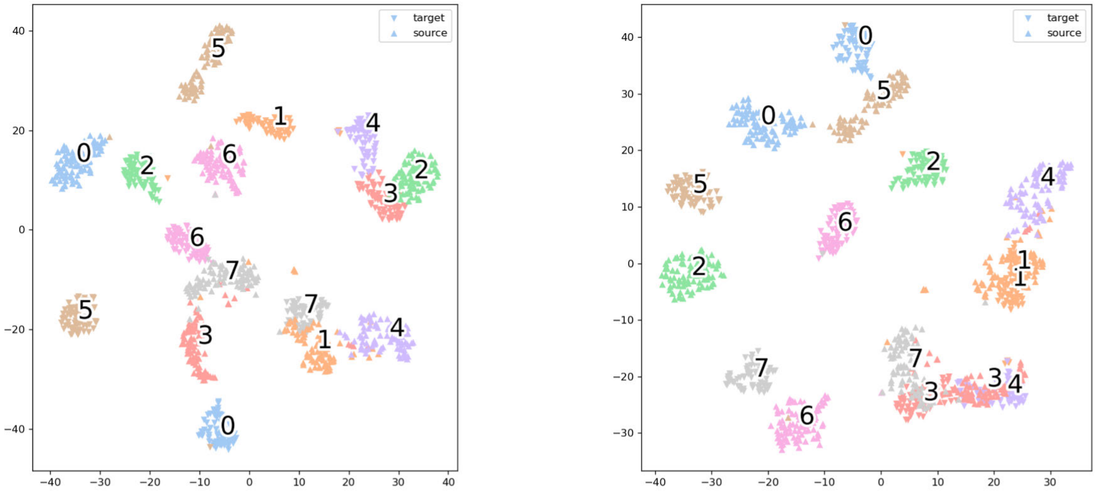
We conduct extensive experiments on two public datasets, NinaPro and CapgMyo. All experimental programs are written in PyCharm 2021.3.2 Professional Edition, compiled with Python 3.8.3, and mainly use torch library, version "1.10.2+cu113". The experiment uses GPU for accelerated training, and the GPU model is NVIDIA RTX A5000 (24GB).

A. Experimental Setup

1) Datasets and Preprocessing: A summary of the dataset is shown in Table I. NinaPro is the most commonly used dataset for sEMG gesture recognition and contains ten sub-datasets. We use two sub-datasets (i.e., DB-1 and DB-5) for detailed

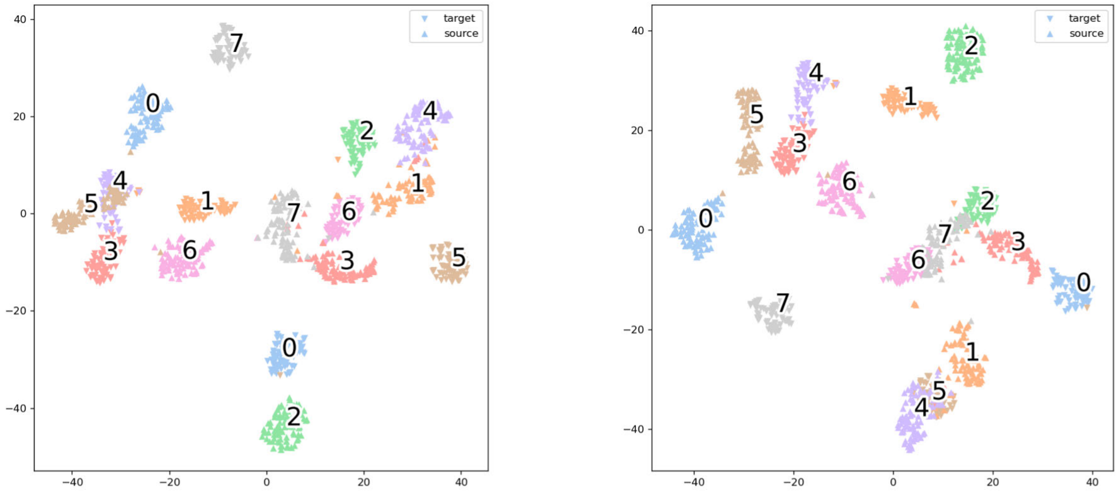
Algorithm 2 The Whole Process of STDA Method

```
Input: Existing user data: $\mathcal{S} = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$, New user
    labeled data: $\mathcal{T}_l = \left\{(x_j^{tl}, y_j^{tl})\right\}_{j=1}^{N_l}$, New user unlabeled
    data: $\mathcal{T}_u = \{(x_k^{tu})\}_{k=1}^{N_u}$, New user test data: $\mathcal{T}_m =$
    $\{(x_m^{te})\}_{m=1}^{N_{te}}$, Parameters of the model: $\Theta$
Output: Prediction results for new user test data: $\{(y_m^{te})\}_{m=1}^{N_{te}}$
repeat
    $clf\_loss \leftarrow \mathcal{S}$, calculated by Formula (3)
    $mmd\_loss \leftarrow \mathcal{S}$ and $\mathcal{T}_u$, calculated by Formula (2)
    $var\_loss \leftarrow \mathcal{T}_u$, calculated by Formula (5)
    $clf\_loss\_2 \leftarrow \mathcal{T}_l$, calculated by Formula (3)
    if $U_{pse} \neq \emptyset$: $clf\_loss\_3 \leftarrow U_{pse}$, calculated by
    Formula (3) else: $clf\_loss\_3 = 0.0$, calculated by
    Algorithm 1
    Calculate total loss
    $loss = clf\_loss + mmd\_loss + var\_loss +$
    $clf\_loss\_2 + clf\_loss\_3$, optimize the parameters by
    Adam
until Iterate until model convergence
Predict the results: $\{(y_m^{te})\}_{m=1}^{N_{te}} \xleftarrow{\Theta} \{(x_m^{te})\}_{m=1}^{N_{te}}$
```



(a) early (b) late

Fig. 7. Visualization of the STDA method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.



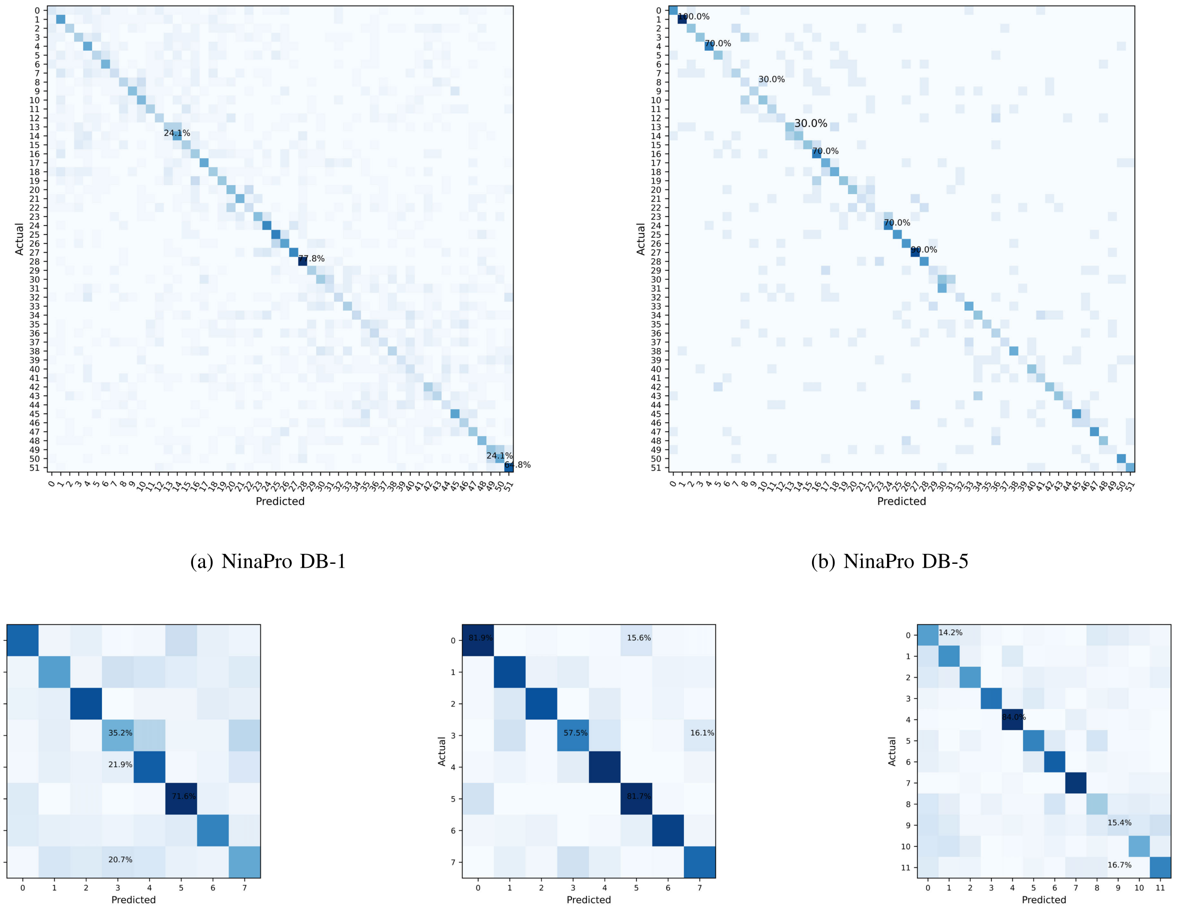
(a) early (b) late

Fig. 8. Visualization of the Multi-Stream' method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colors represent different categories, and there are eight categories in total.

experiments. The sEMG data of DB-1 and DB-5 are collected by sparse electrode sensing devices. The DB-1 dataset contains ten channels, and the DB-5 contains 16 channels. The DB-1 sub-dataset includes 27 healthy subjects with a total of 52 gestures. Among them, 12 basic finger movements were collected in the first collection; eight basic hand movements and nine basic wrist movements were collected in the second collection; grasping and functional movements were collected in the third collection, with 23 gestures. Each gesture repeats ten times. The sampling frequency of DB-1 is 100Hz. The DB-5 dataset's sampling frequency is 200Hz, and the subject number is ten. The number and type of gestures are the same as those of DB-1, but each gesture only repeats six times. The dataset can be obtained from the following URL: http://ninapro.hevs.ch/.

CapgMyo is sEMG data collected by a high-density electrode array sensing device. One hundred twenty-eight channels of high-density sEMG data were collected from 23 healthy subjects at a sampling frequency of 1000 Hz. This dataset includes three sub-datasets, namely DB-a, DB-b and DB-c. DB-a contains eight finger gestures from 18 subjects; DB-b contains eight gestures from 10 subjects collected twice in two different periods; Db-c contains 12 basic finger gestures from 10 subjects. The eight gestures for DB-a and DB-b are from Nos. 13-20 in NinaPro. The 12 gestures in DB-c have derived from basic finger movements Nos. 1-12 in NinaPro. The dataset can be obtained from the following URL: http://zju-capg.org.

In order to obtain more helpful information, the original one-dimensional sEMG time-series signals are processed by the fast Fourier transform based on the Hann window. This transform method can convert the raw sEMG signal to spectral form, i.e., from $\mathbb{R}^{(n,c,h)}$ to $\mathbb{R}^{(n,t,c,f)}$, where n represents the number of samples, c represents the number of channels of sEMG data, h represents the length of the original sEMG signal, t represents time, and f represents frequency.



(a) NinaPro DB-1

(b) NinaPro DB-5

(c) CapgMyo DB-a

(d) CapgMyo DB-b

(e) CapgMyo DB-c

Fig. 9. Confusion matrix analysis.

TABLE I

SUMMARY OF THE EXPERIMENTAL DATASET

Dataset	**#subject**	**#gesture**	**frequency (Hz)**	**#channel**	**#repetion**
DB-1	27	52	100	10	10
DB-5	10	52	200	16	6
DB-a	18	8	1000	128	6
DB-b	20	8	1000	128	6
DB-c	9*	12	1000	128	6

*: Due to server failure, we only got data for the first nine users.

Here, we emphasize our division of the target domain. We take the dataset DB-a as an example. DB-a includes 18 users, each with eight gestures, and each gesture is repeated ten times. The target user (one user) is first split into two parts according to the ratio of 8:2. Specifically, the eight gesture data of the 9th and 10th repetitions are divided into the test part, T_{te}, and the remaining part is called the training part. We removed all the data labels in the training part and assigned them to T_u. We divide the training part into T_l after taking a labeled sample for each gesture, and there are only eight samples in total.

2) Parameter Settings: As the incorrect pseudo-labels will lead to significant deviations in the model's learning process, we set the model's confidence parameter in the experiment as 0.99. First, by utilizing the uncertainty of the deep learning model, we make two consecutive pseudo-label sets for the same data set with a confidence level of 0.99. Then, the two sets of sets are intersected to reduce the number of erroneous pseudo-labels further. Practice in deep learning shows that the pre-trained models perform better and converge faster. This experiment uses the source domain data to pre-train the model for 400 epochs. Using the balance property of the dataset itself, each batch on the target domain is set to a multiple of the total number of gestures in the current dataset. For example, on an 8-category dataset such as DB-a, the batch is set to 48 (8×6) or 56 (8×7), etc. For all datasets, we use leave-one-out cross-validation. In other words, each user is regarded as the target domain once, and all other users are regarded as the source domain. The Adam optimizer optimizes the parameters, and the learning rate is 0.001.

3) Comparison Methods: We compared the STDA method with seven methods. These include two baseline methods, one deep learning method based on fine-tuning, two sEMG gesture recognition methods based on domain adaptation, and two unsupervised domain adaptation methods.

- Only-Source: This is a baseline method. A variant of STDA that uses only the source domain to train the model.
- Only-Target: This is a baseline method. A variant of STDA that uses only the target domain to train the model.
- Multi-Stream': This is a fine-tuning method. First, the raw multi-channel surface EMG signal is decomposed into multiple equal-sized blocks, and each block is called

a stream. Second, each stream learns features through a convolutional neural network. Finally, all the features are fused to train a classifier. We borrow its architecture to train on the source domain data and fine-tune it by adding a small amount of data from the target domain [45].

- MDSDA: This is a method of supervised domain adaptation. A Two-Stream Supervised Domain Adaptation Architecture [35]. Specifically, MDSDA is a CNN-based two-stream architecture. Each stream mainly comprises a convolutional layer, a BN layer, a ReLU layer, a MaxPool2d layer, and a fully connected layer. It is divided into two networks, the source network and the target network. They are structurally consistent but computationally independent of them. The classification loss on both the source and target domains is optimized using the cross-entropy function, and the domain variance loss between the two domains is optimized using the maximum mean discrepancy.
- SGAS: This is a method of unsupervised domain adaptation. A domain adaptation method based on kernel space distance [49]. Specifically, SGAS only screens out data pairs beneficial to the model to update the model, using reliable sample pairs so that the classifier can correctly align the source domain and target domain data distribution.
- Self-Tuning: This is a method for unsupervised domain adaptation. An unsupervised domain adaptation method with a pseudo-group contrast mechanism [43]. Specifically, a pseudo-group contrast mechanism is proposed in Self-Tuning, which reduces the dependence on pseudo-labels and improves the tolerance to wrong labels. It is an effective mechanism to solve the challenge of confirmation bias in self-training.
- CST: This is a method for unsupervised domain adaptation. A recurrent self-training domain adaptation method [23]. More specifically, CST uses the classifier trained in the source domain to generate target domain pseudo-labels, then uses the target domain pseudo-labels to train the target domain classifier, and finally updates the shared representation to make the target domain classifier perform better on the source domain data.

B. Comparative Experiment Results

Using accuracy as an evaluation metric for classification, we evaluate the STDA method. The accuracy rates of STDA and the other seven methods are shown in Table II.

Overall, our STDA method is effective. The highest recognition accuracies on DB-5, DB-a, DB-b and DB-c datasets are 52.69%, 76.31%, 79.86% and 60.44%, respectively. Multi-Stream' is an improved method based on the referenced article. The original multi-stream convolutional network is a pure sEMG gesture recognition method. To evaluate the effectiveness of our approach, we compared it with eight methods. Overall, our method outperforms other methods. Compared with the baseline methods, there is an improvement of more than 25%. Compared with the fine-tuning techniques, there is an improvement of more than 8%, except for the DB-1 dataset. Compared with the supervised domain adaptation methods, there is an improvement of more than 5%. There is more than a 24% improvement compared with the unsupervised domain adaptation methods.

TABLE II
Comparison of Average Accuracy Rates on All Datasets

method	DB-1	DB-5	DB-a	DB-b	DB-c
Only-Source	14.92%	18.46%	21.10%	46.94%	17.28%
Only-Target	2.88%	4.23%	13.34%	14.48%	9.00%
Multi-Stream'	52.38%	44.41%	36.68%	55.10%	32.50%
MDSDA	34.36%	23.07%	37.61%	56.38%	38.62%
SGAS	18.69%	21.34%	23.26%	44.99%	16.46%
Self-Tuning	16.13%	27.69%	23.41%	31.63%	19.13%
CST	2.06%	19.65%	13.58%	12.46%	8.12%
STDA	40.24%	52.69%	76.31%	79.86%	60.44%

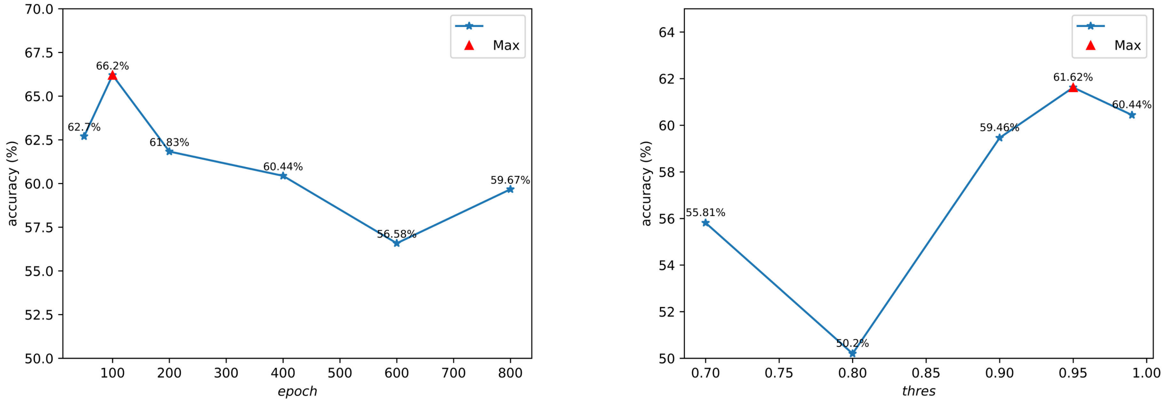
C. Visualization of Distribution changes

To evaluate the performance of the STDA method, we visualize the data distribution on the source and target domains during training using T-distributed random neighbour embeddings. As shown in Fig 7 and Fig 8, we show the comparison between the Multi-Stream' method and our STDA method during the training process and visualize the data distribution of the source domain and the target domain with a comparable amount of data (one user's data). The experiment was carried out on the data set DB-a, the left picture (a) is the early stage of training, and the right picture (b) is the later stage of training. Different colours represent different categories, and there are eight categories in total. The lower triangles represent samples on the target domain, and the upper triangles represent samples on the source domain. Compared with the Multi-Stream' method, our STDA method does have a closer effect on the distribution distance of different domains in the later stage. Visualizations of the rest of the methods are given in the appendix.

D. Confusion Matrix Analysis

Furthermore, we performed a confusion matrix analysis on five datasets. The analysis methods are shown in Fig. 9. From the confusion matrices on the DB-1 and DB-5 datasets, and it can be seen that the 28th and the 51st gestures on DB-1 obtain higher accuracy than other gestures; the 1st, 4th, 16th, 24th and 27th gestures on DB-5 obtained higher accuracy than other gestures, indicating that in the construction of gesture recognition system, an elaborate gesture set design is also crucially important. In addition, it is easy to confuse some gestures. For example, on the DB-1 dataset, the 50th gesture has about a 24% probability of being misjudged as the 49th gesture. Similarly, the 14th gesture is also easily misjudged into the 13th gesture. An analogous phenomenon also occurs on the DB-5 dataset. For example, the 8th gesture is easily misjudged as the 10th gesture, and the 13th gesture is easily misjudged as the 14th gesture. The false positive rate is as high as 30%. Similar conclusions are also obtained on CapgMyo's three sub-datasets, i.e., DB-a, DB-b and DB-c.

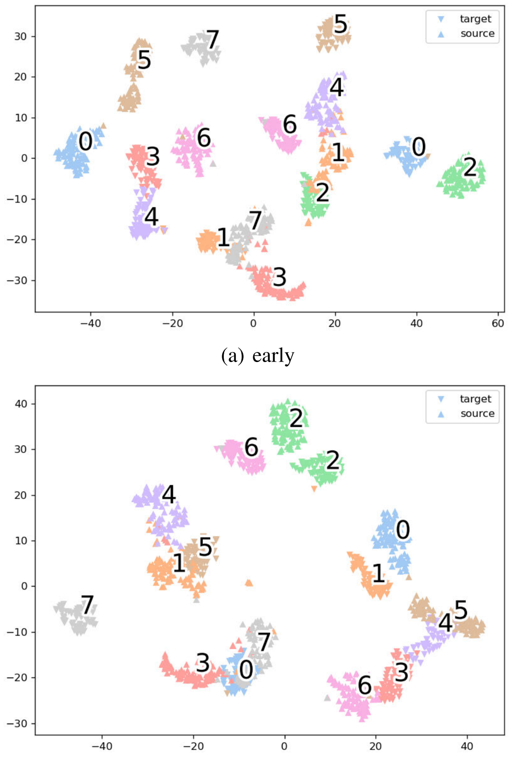
As can be seen from the two 8-class datasets,DB-a and DB-b of the CapgMyo dataset, the fifth gesture achieved the highest accuracy rate of 71.6% in DB-a, the fifth gesture achieved the second highest accuracy, which is very close to



(a) parameter: epoch

(b) parameter: thres

Fig. 10. Parameter sensitivity analysis.



(a) early

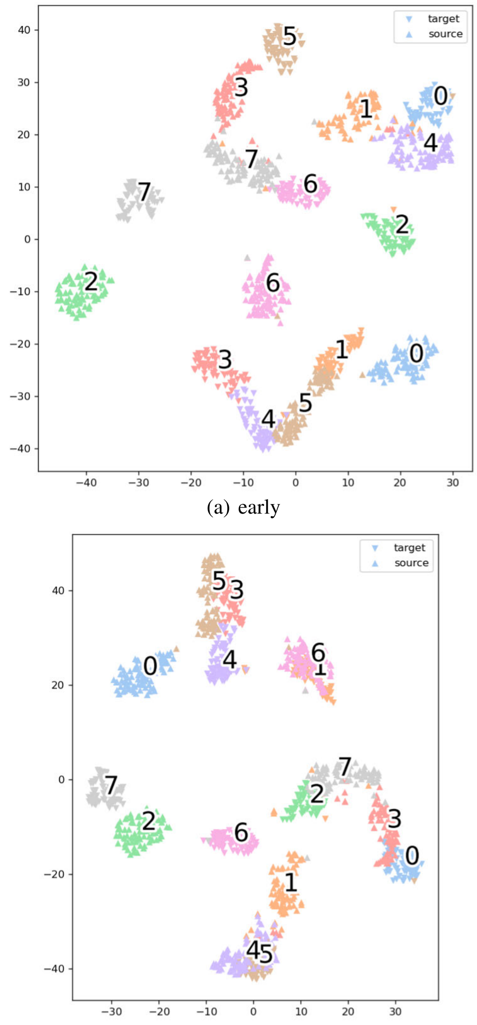
(b) late

Fig. 11. Visualization of the Only-Source method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.

the highest accuracy of 81.7% in DB-b. Meanwhile, the third gesture had the lowest accuracy in DB-a and DB-b, with 35.2% accuracy on DB-a and 57.5% on DB-b. One possible reason is that the fifth gesture is relatively different from the other seven gestures, and the third gesture is very similar to several of the remaining seven gestures.

E. Ablation Experiment Analysis

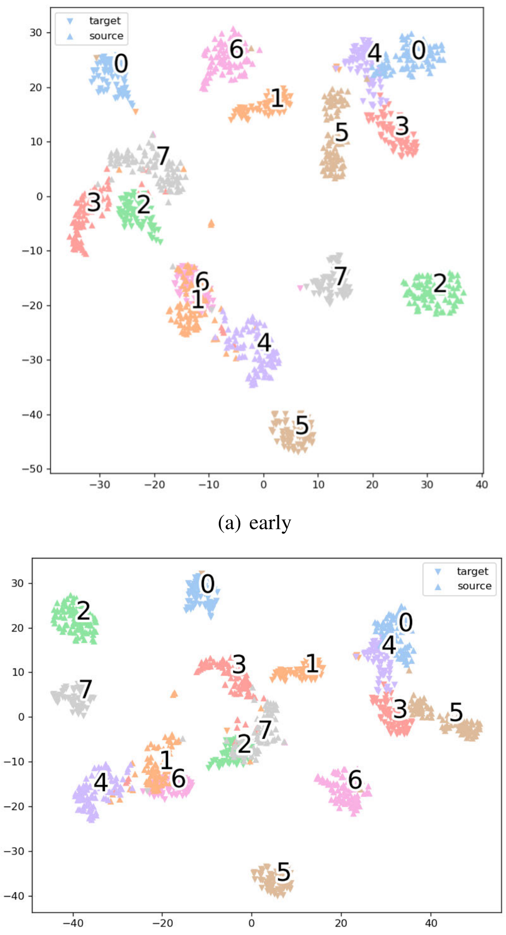
To explore the contribution of each part, we performed ablation experiments. The SDTA method is mainly composed of two modules. The alignment of the feature space is referred to as mmd, and the iterative update of the pseudo-label is referred to as self-training. Experiments on five datasets are shown in Table III.



(a) early

(b) late

Fig. 12. Visualization of the Only-Target method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.



(a) early

(b) late

Fig. 13. Visualization of the MDSDA method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.

TABLE III

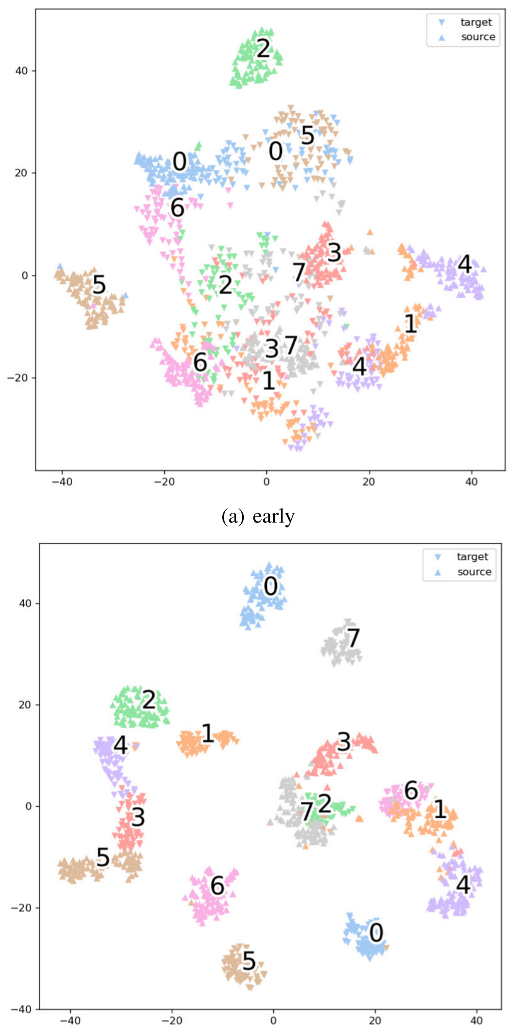
ABLATION ON ALL DATASETS

Method	DB-1	DB-5	DB-a	DB-b	DB-c
STDA w/o mmd	23.78%	51.73%	75.73%	73.78%	60.18%
STDA w/o self-traing	36.00%	47.11%	72.80%	67.39%	43.41%
STDA	40.24%	52.69%	76.31%	79.86%	60.44%

In general, the STDA method combining feature alignment and self-training achieves the best performance, demonstrating our method's effectiveness. At the same time, the experimental results show that the iterative self-training strategy has the most outstanding contribution to DB-5, DB-a, DB-b and DB-c datasets. On the DB-1 dataset, the significant contribution is the alignment of the feature space.

F. Parameter Sensitivity Analysis

The SDTA method is mainly sensitive to two parameters, the number of pre-training epochs (abbreviated as "epoch") and the confidence threshold of iterative self-training (abbreviated as "thres"). To evaluate the effect of parameters on the performance of the STDA method, we used a univariate method, in other words, changing one variable while keeping the other constant. The experimental results are shown in Fig. 10. The range of the parameter epoch is set to {50, 100, 200, 400, 600, 800}, as shown in Fig. 10(a). The range of parameter threshold is set to {0.7, 0.8, 0.9, 0.95, 0.99}, as shown in Fig. 10(b). The red triangle in the figure is the optimal value. From Fig. 10(a), it can be seen that when the *epoch* is 100, the STDA method performs best. Moreover, when the *epoch* is 600, the performance is the worst. Proper pre-training is beneficial to model learning, and excessive pre-training may lead to model overfitting on source domain data. It can be seen from Fig. 10(b) that when the *thres* is 0.95, the STDA method achieves the best performance, and when the *thres* is 0.8, the STDA method has the worst performance. A low confidence level can lead to many false labels, misleading the learning of the model, and overconfidence can also adversely affect the model.



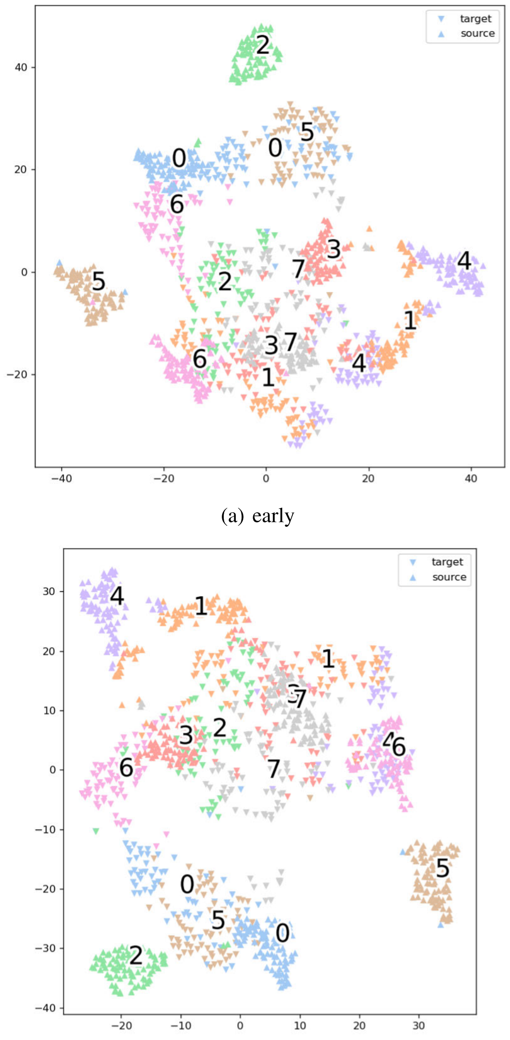
(a) early

(b) late

Fig. 14. Visualization of the SGAS method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.

VII. CONCLUSION AND FUTURE WORKS

This paper proposes an Iterative Self-Training-based Domain Adaptation method (STDA) to solve the cross-user problem. Our motivation is to address the distribution inconsistency of sEMG signal among different subjects through



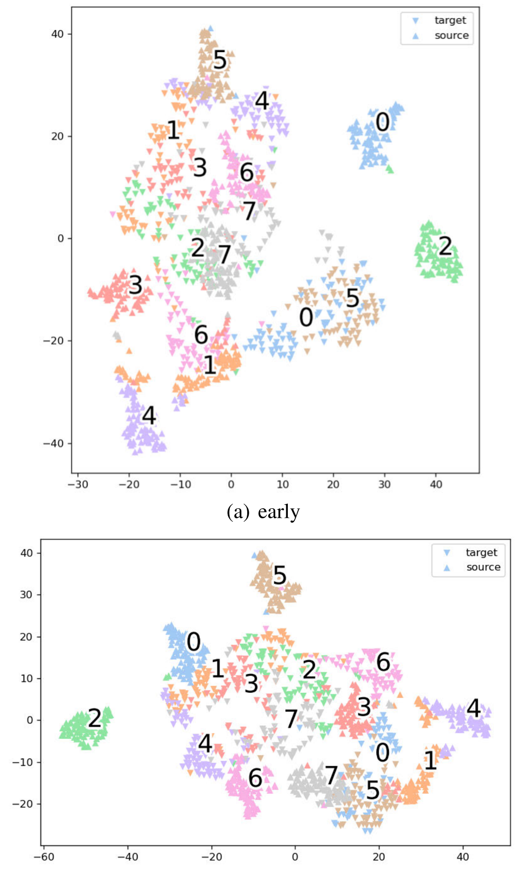
(a) early

(b) late

Fig. 15. Visualization of the Self-Tuning method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.

domain adaptation. The STDA method mainly consists of two parts, discrepancy-based domain adaptation and pseudo-label iterative update. Discrepancy-based domain adaptation shortens the distance between domains through distance constraints in high-dimensional space. Pseudo-label iterative update iteratively generates more accurate pseudo-labels. Apart from that, We compared with the current state-of-the-art methods on five sEMG data sets, and the experimental results showed the progressive nature of our method. Meanwhile, the experimental results of confusion matrix analysis, ablation experiment and parameter sensitivity analysis show the role played by DDA and PIU and the effectiveness of our approach.

However, our method still requires very little labelled data, and the recognition accuracy must be improved before it can be applied to practical systems. In the future, we plan to explore other methods for unsupervised domain adaptation to further reduce the user training burden. Furthermore, we will explore methods for domain generalization to avoid the training burden for new users completely.



(a) early

(b) late

Fig. 16. Visualization of the CST method early and late in the training process (using T-SNE). The upward triangle represents the source domain, and the downward triangle represents the target domain; different colours represent different categories, and there are eight categories in total.

Appendix

A. Early and Late Visualization During Model Training for Other Methods (including Only-Source, Only-Target, MDSDA, SGAS, Self-Tuning and CST)

See Figs. 11–16.

References

[1] O. O. Adebayo, E. Adetiba, and O. T. Ajayi, "Hand gesture recognition-based control of motorized wheelchair using electromyography sensors and recurrent neural network," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 1107, no. 1, Apr. 2021, Art. no. 012063.

[2] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review," *Remote Sens.*, vol. 13, no. 3, p. 527, Feb. 2021.

[3] D. Avola, L. Cinque, A. Fagioli, G. L. Foresti, A. Fragomeni, and D. Pannone, "3D hand pose and shape estimation from RGB images for keypoint-based hand gesture recognition," *Pattern Recognit.*, vol. 129, Sep. 2022, Art. no. 108762.

[4] G. Ayana, J. Park, J.-W. Jeong, and S.-W. Choe, "A novel multistage transfer learning for ultrasound breast cancer image classification," *Diagnostics*, vol. 12, no. 1, p. 135, Jan. 2022.

[5] D. Bai, T. Liu, X. Han, and H. Yi, "Application research on optimization algorithm of sEMG gesture recognition based on light CNN+LSTM model," *Cyborg Bionic Syst.*, vol. 2021, Jan. 2021, Art. no. 9794610.

[6] H. Bi, M. Perello-Nieto, R. Santos-Rodriguez, P. Flach, and I. Craddock, "An active semi-supervised deep learning model for human activity recognition," *J. Ambient Intell. Hum. Comput.*, pp. 1–17, Mar. 2022.

[7] E. Campbell, A. Phinyomark, and E. Scheme, "Deep cross-user models reduce the training burden in myoelectric control," *Frontiers Neurosci.*, vol. 15, p. 595, May 2021.

[8] P. P. K. Chan et al., "Unsupervised domain adaptation for gesture identification against electrode shift," *IEEE Trans. Human-Mach. Syst.*, vol. 52, no. 6, pp. 1271–1280, Dec. 2022.

[9] B. A. De la Cruz-Sánchez, M. Arias-Montiel, and E. Lugo-González, "EMG-controlled hand exoskeleton for assisted bilateral rehabilitation," *Biocybern. Biomed. Eng.*, vol. 42, no. 2, pp. 596–614, Apr. 2022.

[10] Y. Du et al., "Semi-supervised learning for surface EMG-based gesture recognition," in *Proc. IJCAI*, Aug. 2017, pp. 1624–1630.

[11] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," in *Advances in Data Science and Information Engineering*. Cham, Switzerland: Springer, 2021, pp. 877–894.

[12] A. Ghorbanali, M. K. Sohrabi, and F. Yaghmaee, "Ensemble transfer learning-based multimodal sentiment analysis using weighted convolutional neural networks," *Inf. Process. Manage.*, vol. 59, no. 3, May 2022, Art. no. 102929.

[13] W. Guo, M. Wang, X. Sheng, and X. Zhu, "Towards semi-supervised myoelectric finger motion recognition based on spatial motor units activation," *Sci. China Technol. Sci.*, vol. 65, pp. 1232–1242, May 2022.

[14] H. Huang et al., "Virtual interaction and manipulation control of a hexacopter through hand gesture recognition from a data glove," *Robotica*, vol. 40, no. 12, pp. 1–13, 2022.

[15] D. Josephs, C. Drake, A. Heroy, and J. Santerre, "sEMG gesture recognition with a simple model of attention," in *Proc. Mach. Learn. Health*, 2020, pp. 126–138.

[16] P. Jung, G. Lim, S. Kim, and K. Kong, "A wearable gesture recognition device for detecting muscular activities based on air-pressure sensors," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 485–494, Apr. 2015.

[17] S. Kanoga, T. Hoshino, and H. Asoh, "Subject-transfer framework with unlabeled data based on multiple distance measures for surface electromyogram pattern recognition," *Biomed. Signal Process. Control*, vol. 74, Apr. 2022, Art. no. 103522.

[18] I. Ketykó, F. Kovács, and K. Z. Varga, "Domain adaptation for sEMG-based gesture recognition with recurrent neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7.

[19] P. Kieliba, D. Clode, R. O. Maimon-Mor, and T. R. Makin, "Robotic hand augmentation drives changes in neural body representation," *Sci. Robot.*, vol. 6, no. 54, May 2021, Art. no. eabd7935.

[20] J. Kim, S. Mastnik, and E. André, "EMG-based hand gesture recognition for realtime biosignal interfacing," in *Proc. 13th Int. Conf. Intell. User Interfaces*, Jan. 2008, pp. 30–39.

[21] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 1, pp. 1–58, Mar. 2009.

[22] C. Li, T. Zhen, and Z. Li, "Image classification of pests with residual neural network based on transfer learning," *Appl. Sci.*, vol. 12, no. 9, p. 4356, Apr. 2022.

[23] H. Liu, J. Wang, and M. Long, "Cycle self-training for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 22968–22981.

[24] X. Lv et al., "Gesture recognition based on sEMG using multi-attention mechanism for remote control," *Neural Comput. Appl.*, vol. 35, pp. 13839–13849, Jul. 2023.

[25] E. Mammen, "Nonparametric regression under qualitative smoothness assumptions," *Ann. Statist.*, vol. 19, no. 2, pp. 741–759, Jun. 1991.

[26] E. D. Oña, S. Marcos-Antón, D.-S. Copaci, J. Arias, R. Cano-de-la-Cuerda, and A. Jardón, "Effects of EMG-controlled video games on the upper limb functionality in patients with multiple sclerosis: A feasibility study and development description," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–16, Apr. 2022.

[27] M. A. Ozdemir, D. H. Kisa, O. Guren, and A. Akan, "Hand gesture classification using time–frequency images and transfer learning based on CNN," *Biomed. Signal Process. Control*, vol. 77, Aug. 2022, Art. no. 103787.

[28] M. A. Ozdemir, D. H. Kisa, O. Guren, A. Onan, and A. Akan, "EMG based hand gesture recognition using deep learning," in *Proc. Med. Technol. Congr. (TIPTEKNO)*, Nov. 2020, pp. 1–4.

[29] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[30] N. J. Prottasha et al., "Transfer learning for sentiment analysis using BERT based supervised fine-tuning," *Sensors*, vol. 22, no. 11, p. 4157, May 2022.

[31] E. Rahimian, S. Zabihi, A. Asif, S. F. Atashzar, and A. Mohammadi, "Trustworthy adaptation with few-shot learning for hand gesture recognition," in *Proc. IEEE Int. Conf. Auto. Syst. (ICAS)*, Aug. 2021, pp. 1–5.

[32] M. Rajagopalaswamy, M. Sivapalanirajan, D. Vairamuthu, S. Prince Joshwa, T. Vigneshkumar, and M. W. Iruthayarajan, "Accelerometer based hand gesture recognition and control of motor powered trolley," in *Proc. 10th Int. Conf. Emerg. Trends Eng. Technol. Signal Inf. Process. (ICETET-SIP)*, Apr. 2022, pp. 1–5.

[33] S. D. Regani, B. Wang, and K. J. R. Liu, "WiFi-based device-free gesture recognition through-the-wall," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8017–8021.

[34] A. Sahito, E. Frank, and B. Pfahringer, "Better self-training for image classification through self-supervision," in *Proc. Australas. Joint Conf. Artif. Intell.* Cham, Switzerland: Springer, 2022, pp. 645–657.

[35] P. Shi, X. Zhang, W. Li, and H. Yu, "Improving the robustness and adaptability of sEMG-based pattern recognition using deep domain adaptation," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 11, pp. 5450–5460, Nov. 2022.

[36] X. Song, S. S. Van De Ven, L. Liu, F. J. Wouda, H. Wang, and P. B. Shull, "Activities of daily living-based rehabilitation system for arm and hand motor function retraining after stroke," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 621–631, 2022.

[37] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "Deep learning in EMG-based gesture recognition," in *Proc. PhyCS*, 2018, pp. 107–114.

[38] F. van Ede and A. C. Nobre, "Turning attention inside out: How working memory serves behavior," *Annu. Rev. Psychol.*, vol. 74, no. 1, pp. 137–165, Jan. 2023.

[39] J. A. van Oijen, A. Donini, R. J. M. Bastiaans, J. H. M. T. T. Boonkkamp, and L. P. H. de Goey, "State-of-the-art in premixed combustion modeling using flamelet generated manifolds," *Prog. Energy Combustion Sci.*, vol. 57, pp. 30–74, Nov. 2016.

[40] H. Wang, Y. Zhang, C. Liu, and H. Liu, "SEMG based hand gesture recognition with deformable convolutional network," *Int. J. Mach. Learn. Cybern.*, vol. 13, no. 6, pp. 1729–1738, Jun. 2022.

[41] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, early access, May 26, 2022, doi: 10.1109/TKDE.2022.3178128.

[42] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[43] X. Wang, J. Gao, M. Long, and J. Wang, "Self-tuning for data-efficient deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10738–10748.

[44] Y. Wang, Y. Tian, H. She, Y. Jiang, H. Yokoi, and Y. Liu, "Design of an effective prosthetic hand system for adaptive grasping with the control of myoelectric pattern recognition approach," *Micromachines*, vol. 13, no. 2, p. 219, Jan. 2022.

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

[46] J. Xu et al., "Adaptive semi-supervised intent inferral to control a powered hand orthosis for stroke," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8097–8103.

[47] X. Yang, F. Lv, F. Liu, and G. Lin, "Self-training vision language BERTs with a unified conditional model," 2022, *arXiv:2201.02010*.

[48] X. Yang, J. Yan, and H. Liu, "Comparative analysis of wearable A-mode ultrasound and sEMG for muscle-computer interface," *IEEE Trans. Biomed. Eng.*, vol. 67, no. 9, pp. 2434–2442, Sep. 2020.

[49] X. Zhang, X. Zhang, L. Wu, C. Li, X. Chen, and X. Chen, "Domain adaptation with self-guided adaptive sampling strategy: Feature alignment for cross-user myoelectric pattern recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1374–1383, 2022.

[50] Y. Zhang, Y. Chen, H. Yu, X. Yang, and W. Lu, "Dual layer transfer learning for sEMG-based user-independent gesture recognition," *Pers. Ubiquitous Comput.*, vol. 26, pp. 575–586, Jun. 2022.

[51] N. Zheng, Y. Li, W. Zhang, and M. Du, "User-independent EMG gesture recognition method based on adaptive learning," *Frontiers Neurosci.*, vol. 16, Mar. 2022, Art. no. 847180.

[52] X. J. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep. 1530, 2005.