

Cross-Subject Lifelong Learning for Continuous Estimation From Surface Electromyographic Signal

Xingjian Chen¹, Weiyu Guo, Chuang Lin², Ning Jiang³, *Senior Member, IEEE*, and Jingyong Su¹

Abstract—The employment of surface electromyographic (sEMG) signals in the estimation of hand kinematics represents a promising non-invasive methodology for the advancement of human-machine interfaces. However, the limitations of existing subject-specific methods are obvious as they confine the application to individual models that are custom-tailored for specific subjects, thereby reducing the potential for broader applicability. In addition, current cross-subject methods are challenged in their ability to simultaneously cater to the needs of both new and existing users effectively. To overcome these challenges, we propose the Cross-Subject Lifelong Network (CSLN). CSLN incorporates a novel lifelong learning approach, maintaining the patterns of sEMG signals across a varied user population and across different temporal scales. Our method enhances the generalization of acquired patterns, making it applicable to various individuals and temporal contexts. Our experimental investigations, encompassing both joint and sequential training approaches, demonstrate that the CSLN model not only attains enhanced performance in cross-subject scenarios but also effectively addresses the issue of catastrophic forgetting, thereby augmenting training efficacy.

Index Terms—sEMG, cross-subject, continuous estimation, lifelong learning, hand kinematics.

I. INTRODUCTION

THE advancements in robotics, intelligent devices, and internet technologies have significantly increased the

Manuscript received 18 February 2024; revised 29 April 2024; accepted 7 May 2024. Date of publication 13 May 2024; date of current version 22 May 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62376068, in part by Guangdong Regional Joint Funds Key Projects under Grant 2023B1515120065, and in part by Shenzhen Science and Technology Innovation Program under Grant JCYJ20220818102414031. (Xingjian Chen and Weiyu Guo contributed equally to this work.) (Corresponding authors: Jingyong Su; Chuang Lin.)

Xingjian Chen and Jingyong Su are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China (e-mail: sujingyong@hit.edu.cn).

Weiyu Guo is with the Department of Artificial Intelligence Thrust, The Hong Kong University of Science and Technology, Hong Kong.

Chuang Lin is with the School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China (e-mail: chuang.lin@dlnu.edu.cn).

Ning Jiang is with the National Clinical Research Center for Geriatrics, West China Hospital, and the Med-X Center for Manufacturing, Sichuan University, Chengdu 610017, China.

Data is available on-line at <https://github.com/Great-P/sEMG-master>.
Digital Object Identifier 10.1109/TNSRE.2024.3400535

demand for efficient Human-Machine Interfaces (HMI), especially in areas like smart homes, industrial automation, and healthcare. User-friendly and high-quality interaction signals are essential in these contexts. Surface electromyography (sEMG) is a promising technology in this regard, playing a critical role in interpreting human intentions from biological signals. Accurate extraction of motion intentions from sEMG signals is crucial in high-precision related tasks of HMI, ensuring reliable and responsive control. The effectiveness of sEMG, as demonstrated in various studies [1], [2], [3], [4], [5], also shows considerable potential in the development of HMI.

Deep learning methods are widely applied in sEMG-based motion estimation tasks. Techniques employing artificial neural networks (ANNs) [6] have been proposed for joint motion estimation, although they are typically limited to specific degrees of freedom. Subsequent advancements introduced methods based on Recurrent Neural Networks (RNNs) for hand pose estimation [7], despite facing computational challenges. Recent developments include the integration of Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks [4], [8], enhancing both performance and efficiency. However, existing methods primarily focus on individual subjects, underscoring the necessity for more universal approaches capable of handling multiple subjects and supporting knowledge transfer among them.

sEMG signals are markedly varied among individuals, presenting substantial challenges in developing versatile models for wide-ranging applications. Subject-specific models excel in certain gesture recognition tasks, yet they are impractical for broader applications, such as those in public facilities and industrial settings. The use of a unique model for each individual, coupled with frequent personnel changes, can lead to substantial costs. Thus, there is a clear need for a singular, adaptable model that ensures seamless integration across various users, devices, and scenarios. This model must be robust and flexible, capable of accommodating the unique characteristics of surface electromyography (sEMG) signals from each user without requiring extensive training.

Recent advancements in cross-subject studies tackle sEMG signal variability through two primary strategies. Transfer learning techniques, such as those described in [9], adapt subject-specific models for a broader user base. For example, Fan et al. [10] developed a hand gesture recognition method

benefiting both intact-handed individuals and amputees. Yet, these methods encounter practical limitations, including increased memory demands and scalability challenges, necessitating distinct models for each user. As an alternative, cross-subject methods have emerged. Our previous research [5] introduced a BERT-based approach with μ -law normalization, enabling effective cross-subject training. Similarly, Long et al. [11] utilized adversarial transfer in a multi-scale model for sEMG signal interpretation across subjects. However, these prevalent transfer learning methods, including those of Long et al. [11], often prioritize new users and struggle to maintain robust predictions for previous users, leading to catastrophic forgetting and an inability to retain information from earlier users. In contrast, self-contained models like our BERT-based approach [5] focus on current training data but face difficulties in preserving knowledge from past users.

Lifelong learning is crucial for addressing catastrophic forgetting in sEMG-based tasks, combining adaptability and knowledge preservation. This method overcomes the limitations of subject-specific models and boosts efficient learning and memory usage. Progressive neural networks [12], [13] have notably tackled catastrophic forgetting in gesture classification. Classic lifelong learning methods like Learning without Forgetting (LwF), Elastic Weight Consolidation (EWC), and Gradient Episodic Memory (GEM) [14] have been applied in sequential time series. Nonetheless, applying lifelong learning to continuous sEMG estimation presents unique challenges, such as complex feature extraction and balancing between preserving existing knowledge and adapting to new users. Research exploring specific lifelong learning strategies for continuous sEMG estimation remains limited.

In this study, we introduce an innovative cross-subject model that incorporates lifelong learning, effectively addressing variability in sEMG-based tasks and demonstrating superior performance. Notably, this research marks the first adaptation of lifelong learning to sEMG-based regression tasks.

Our contributions are summarized as follows:

- We propose a cross-subject model that excels in continuous sEMG prediction.
- Our model incorporates a novel lifelong learning strategy within the CSLN, which rapidly adapts to new users while retaining knowledge from previous users.
- Our method achieves state-of-the-art performance in hand kinematics estimation, surpassing existing methods not only in cross-subject settings but also in lifelong learning scenarios.

II. RELATED WORK

A. Transfer Learning

Transfer learning [9] is a paradigm in artificial intelligence where knowledge gained from solving one problem is applied to a different but related problem. This approach has garnered significant attention due to its ability to leverage pre-existing knowledge and adapt it to new tasks, often leading to improved performance or efficiency, especially when labeled data for the target task is limited. By transferring knowledge learned from a source domain to a target domain, transfer learning

enables models to generalize better and requires less training data, making it particularly useful in scenarios where data collection is expensive or time-consuming. Fine-tuning and Domain Adaptation are two widely used transfer learning methods.

B. Lifelong Learning

Lifelong learning, alternatively known as continual or incremental learning, has risen as a key approach to counter catastrophic forgetting, a phenomenon causing models to underperform or fail with changing data distributions. This leads to a rapid decline in performance on previously learned tasks. Recent research in this field, exemplified by works such as [15], [16], and [17], can be categorized into four main strategies [15]: gradient-based, modularity, memory-based, and meta-learning methods.

Gradient-based methods focus on modulating task gradients, utilizing techniques such as gradient alignment [18], [19], regularization [20], and knowledge distillation [21], [22] to mitigate forgetting.

Modularity methods aim to balance between a single network that is prone to forgetting and multiple networks that impede task transfer. These methods adapt by adding new parameters for new tasks [23], [24] or implementing sparsity strategies [25], [26].

Memory-based methods excel in long-term knowledge retention, typically involving replay or rehearsal techniques [27], [28], [29], [30] and generative memory models [31], [32], [33].

Finally, meta-learning approaches [34], [35] focus on automatically learning inductive biases related to architecture, data, and learning parameters, which traditionally required manual design.

III. CROSS-SUBJECT MODEL WITH LIFELONG LEARNING

A. Data Preprocessing

In this study, the Root Mean Square (RMS) feature is extracted utilizing a sliding window of 100 ms with a 0.5 ms step, as per the long exposure method [4]. This window size and step are chosen to balance temporal resolution and computational efficiency, effectively reducing noise and revealing more sequential features in the data. Following feature extraction, a logarithmic normalization, specifically μ -law normalization [36], [37], is applied to standardize the extracted features. The normalization can be formulated as:

$$F(x_t) = \text{sign}(x_t) \frac{\ln(1 + \mu|x_t|)}{\ln(1 + \mu)} \quad (1)$$

Here, x_t denotes the input raw data, and μ is a hyperparameter, set to $\mu = 2^{20}$ in our study. The choice of μ is based on an exploration range from 2^5 to 2^{25} , optimizing for the best performance in data normalization.

B. Cross-Subject Lifelong Network

The Cross-Subject Lifelong Network (CSLN) features an encoder-decoder architecture with two main branches: a regression branch and a classification branch. In the regression branch, a Temporal Convolutional Network (TCN) based

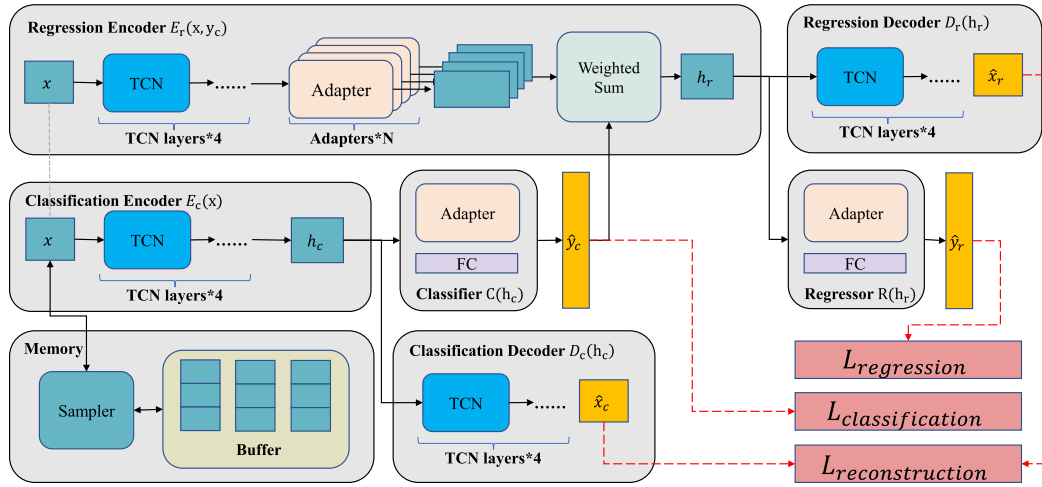


Fig. 1. CSLN Model Architecture: A Dual-Branch Cross-Subject Model for sEMG-Based Hand Motion Prediction with Enhanced Knowledge Retention. The regression branch is responsible for predicting motion based on learned knowledge, while the classification branch functions to select relevant knowledge. Common knowledge is stored by the TCN and regressor in the regression branch, whereas individual knowledge is stored in the adapters.

encoder extracts shared knowledge from sEMG signals, while individual features are generated through adapters. The classification branch, also employing a TCN, predicts subject labels, aiding in selecting specific knowledge for the regression branch. Two decoders then reconstruct the input signal using these features, enhancing cross-subject task performance. Figure 1 shows the CSLN structure.

The final phase of CSLN involves regression and classification tasks to predict hand motion and classify subjects. Adapters and linear layers with distinct parameters are used for adapting to new tasks, maximizing the utility of extracted features.

Adapters, as defined by Hously et al. [38], are compact neural network modules placed between pre-trained base layers and task-specific layers. They efficiently adapt models to new tasks or domains by learning limited task-specific parameters. In CSLN, adapters play a crucial role in acquiring subject-specific parameters, supporting the adaptation to various subjects and the lifelong learning process. Figure 2 displays the adapter structure used in CSLN, with the adapter function $\text{Ada}(x)$ specified as follows:

$$\begin{aligned} x_{down} &= \text{FC}_{(c,c/2)}(x)x_{nl} = \text{GELU}(x_{down}) \\ \text{output} &= \text{FC}_{(c/2,c)}(x_{nl}) + x \end{aligned} \quad (2)$$

where GELU represents the Gaussian Error Linear Unit activation function and $\text{FC}_{(c_{in},c_{out})}$ denotes a linear layer with input channel c_{in} and output channel c_{out} .

Several key components are essential to outline the model's procedural steps. The input sEMG signal is denoted as $x \in \mathcal{R}^{c \times t}$, with c representing the number of channels and t the window size. Regression and classification tasks use encoders E_r and E_c , and their corresponding decoders are D_r and D_c . The outputs include the classification soft label \hat{y}_c , motion prediction \hat{y}_r , and reconstruction \hat{x} . The overall logical procedure is:

$$\begin{aligned} h_c &= E_c(x) & \hat{y}_c &= C(h_c) & h_r &= E_r(x, \hat{y}_c) \\ \hat{y}_r &= R(h_r) & \hat{x}_r &= D_r(h_r) & \hat{x}_c &= D_c(h_c) \end{aligned} \quad (3)$$

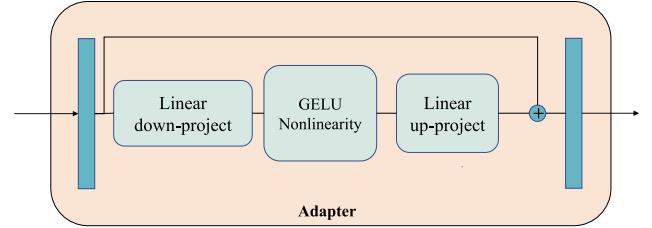


Fig. 2. The Lightweight and Modular Adapter Architecture in CSLN for Knowledge Retention.

where the h_r and h_c represent the features extracted for regression and classification, respectively. In order to describe the process, we define the process $\text{TCN}_{(c_i, c_o)}(\mathbf{I})$ as follows:

$$\begin{aligned} o_1 &= \mathcal{F}_{(c_i, c_o)}^1(\mathbf{I}) + \mathbf{I} & o_2 &= \mathcal{F}_{(c_o, c_o)}^2(o_1) + o_1 \\ o_3 &= \mathcal{F}_{(c_o, c_o)}^4(o_2) + o_2 & \mathbf{O} &= \mathcal{F}_{(c_o, c_o)}^8(o_3) + o_3 \end{aligned} \quad (4)$$

Here, \mathbf{I} and \mathbf{O} represent the input and output, c_i, c_o represents the input and output channel of the whole $\text{TCN}_{(c_i, c_o)}$ process, $\mathcal{F}_{(c_{in}, c_{out})}^d$ refers to the causal and dilated convolutions used in TCN [39], where d is the dilation rate, and c_{in} and c_{out} are the input and output channels of \mathcal{F} , respectively.

Then, the operation of $h_c = E_c(x)$, $h_r = E_r(x, \hat{y}_c)$ and $\hat{x} = D(h)$ is further detailed as follows:

$$\begin{aligned} h_c &= E_c(x) = \text{TCN}_{(c, c_h)}(x) \\ a_i &= \text{Ada}_i(\text{TCN}_{(c, c_h)}(x)) \\ h_r &= E_r(x, \hat{y}_c) = \hat{y}_c^T \mathbf{A} \\ \hat{x} &= D(h) = \text{TCN}_{(c_h, c)}(h) \end{aligned} \quad (5)$$

where c signifies the channel number for the input RMS feature of sEMG, and c_h represents the hidden feature channel in our proposed model. N represents the total number of subjects among all trained tasks. The vector $\mathbf{A} = \{a_1, a_2, \dots, a_n\}$, n represents the number of subjects in all the previous tasks. \hat{x} is the reconstructed sEMG signal. The decoder is divided into two parts: D_r for regression and D_c for classification, operating on the respective hidden features h_r and h_c .

Linear layers, in conjunction with adapters, function as classifiers and regressors in our model, enabling the inference and

output generation from hidden features, the final processing steps are summarized as:

$$\begin{aligned}\hat{y}_c &= C(h_c) = \text{Softmax}(FC_{(c,N)}(\text{Adapter}(h_c))) \\ \hat{y}_r &= R(h_r) = FC_{(c,c_o)}(\text{Adapter}(h_r))\end{aligned}\quad (6)$$

Here, C and R denote the classifier and regressor, respectively, where N is the number of subjects and c_o signifies the output channels, with $c_o = 10$ in this study.

The loss function is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_r + \mathcal{L}_{rec} + \lambda \mathcal{L}_c \quad (7)$$

In this equation, \mathcal{L}_r , \mathcal{L}_{rec} , and \mathcal{L}_c represent the regression loss, reconstruction loss, and classification loss, respectively. The hyperparameter λ is employed to balance these components.

Specifically, the loss functions are defined as:

$$\begin{aligned}\mathcal{L}_r &= \text{MSE}(\hat{y}_r, y_r) & \mathcal{L}_c &= \text{CE}(\hat{y}_c, y_c) \\ \mathcal{L}_{rec} &= \text{MSE}(\hat{x}_r, x) + \text{MSE}(\hat{x}_c, x)\end{aligned}\quad (8)$$

Here, MSE refers to the mean square error loss and CE to the cross-entropy loss. The terms y_r and y_c are the ground truth labels for hand motion and subject classification, respectively.

C. Lifelong Learning Strategy

While the Cross-Subject Lifelong Network (CSLN) demonstrates proficiency in cross-subject scenarios, a challenge arises in generalizing to new users without compromising the knowledge acquired from previous users. To tackle this issue, we integrate a lifelong learning strategy into CSLN.

Inspired by existing research in transfer and lifelong learning [33], [40], our approach employs a hybrid strategy that combines modularity with rehearsal. This method utilizes the modularity concept to transform the lifelong learning regression challenge into a more manageable classification problem, which is then addressed using a rehearsal technique.

For the regression branch, we employ a modularity method [15], adopting the adapter structure commonly used in transfer learning for fields such as natural language processing [38] and computer vision [41]. This structure facilitates efficient parameter expansion and retraining for new tasks.

The classification branch aids in selecting knowledge via the outputs of multi-head adapters. These adapters produce feature outputs that are aggregated to form a comprehensive feature vector. The vector, when multiplied by individual classification labels, produces the regression feature, enabling the network to learn continually from new users while preventing catastrophic forgetting. For transferring knowledge in the classification branch, we implement a replay strategy, incorporating a memory module.

The memory module comprises a storage buffer and a sampler. The buffer stores observed RMS features with associated subject labels, assigned sequentially as array indices for efficient memory usage. It ensures equal capacity for the data from each subject. The sampler uniformly selects inputs from all subjects to maintain consistent batch sizes. After training, it randomly adds new observations to the buffer, supporting continuous learning and effective memory management.

The lifelong learning process in our model is delineated into three critical phases:

(1) **Preparing Phase:** Before embarking on new tasks, the model undergoes an expansion process. This involves integrating new adapters and inheriting parameters from previously used adapters linked to user classification. To accommodate an increasing number of subjects across all tasks, we expand the linear layer in the classifier C accordingly.

(2) **Training Phase:** In this phase, the model trains on new tasks. A sampler effectively merges input data from new subjects with replay data proportionately, based on the number of subjects. Due to the absence of ground truth for hand motion in old data, we modify the loss function to:

$$\mathcal{L}_{lifelong} = \mathcal{L}_r + \mathcal{L}'_{rec} + \lambda \mathcal{L}'_c \quad (9)$$

Here, \mathcal{L}' signifies the loss derived from uniformly sampled data, whereas \mathcal{L} pertains to the loss from solely new data. This equation aims to balance the input from different data sources in the lifelong learning framework. The parameter λ , which is pivotal for the role classification plays in maintaining knowledge of previously trained subjects, is set to 1×10^4 in our study.

(3) **Storing Phase:** Post-training, the sampler randomly selects an equal number of observations for each new subject, allocating them to the memory buffer. In our approach, 100 observations per subject are stored in the buffer.

IV. EXPERIMENTS

A. Dataset

The Ninapro dataset [42], a widely recognized resource in the field of sEMG, provides comprehensive data covering both intact and amputated hands, encompassing over 300 acquisitions from 10 datasets, provides a diverse range of electromyography and kinematics data. This dataset is notable for its extensive collection of sEMG signals captured using the Delsys Trigno Wireless System with 12 electrodes. Additionally, it includes hand kinematics data measured by 22 joint angles via CyberGlove II data gloves. The sEMG data is sampled at a frequency of 2 kHz, while the hand kinematics, initially recorded at 20 Hz, are resampled to match the 2 kHz sEMG data. As such, the Ninapro dataset is an invaluable benchmark for evaluating sEMG-based human-machine interfaces.

In this study, focus is placed on six hand movements related to grasping, targeting 10 finger joints to ensure data quality. All 40 subjects from DB2 are included in cross-subject settings, showcasing our structure's effectiveness. In all settings, 7/10 of each subject is used for training and 3/10 for testing. In lifelong learning experiments, these subjects are divided into 5 tasks (t_i , where $i \in 1, \dots, 5$), with sequential training on each task. To facilitate a thorough evaluation and address potential biases, 5 varied task sequences are generated, detailed in Table I.

B. Evaluation Criteria

To rigorously assess and benchmark our method against other models, we utilize several key evaluation criteria:

1) **Pearson Correlation Coefficient (CC):** The Pearson Correlation Coefficient (CC) is a commonly employed metric that quantifies the linear relationship between two variables. Ranging from -1 to 1 , a higher CC value indicates a stronger

TABLE I

THE DETAIL OF ALL THE TASK SEQUENCES. THE NUMBERS IN THE TABLE REPRESENT THE NUMBER OF SUBJECTS IN NINAPRO DB2. S_x REPRESENTS A TASK SEQUENCE, WHILE T_x REPRESENTS A TASK

Sequence	T_1	T_2	T_3	T_4	T_5
S_1	8,2,39,36,19,35	16,13,27,24,32,12,40,26,25,11,18	34,38,30,20,37,4	21,1,7,22,3,28,5,23,31	15,9,6,10,17,33,14,29
S_2	37,29,24,20,36,23,13,31,28	12,26,34,19,35,15,27,25,22,14,11,16	39,1,30,6,3	40,4,5,9,18,7,32	33,17,8,2,38,10,21
S_3	37,23,7,30,6,18,21,36	16,14,38,1,3	40,8,26,31,39,12,25	11,27,5,9,10,15,17,28,19,35,29	4,13,22,34,32,33,20,24,2
S_4	10,25,15,32,31,33,4,40,6,23	17,36,9,3,29,35,1,7,22,5	18,13,26,37,34,14,30	27,16,38,8	21,19,2,11,20,28,24,39,12
S_5	9,21,26,20,35,5,18,7,32,31	24,22,25,27,1,19,17,4,37	10,34,11,14,33,8,38,15,23,6	12,16,29,39,40,13,2,28	30,36,3

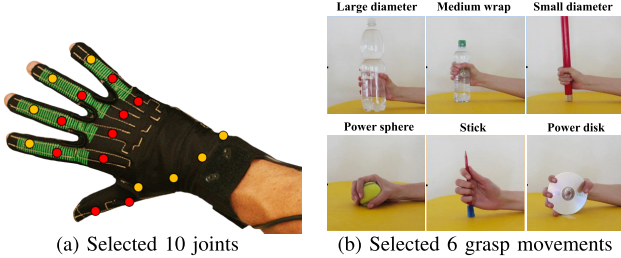


Fig. 3. Data description for this work. CyberGlove channels are shown in (a), where the red dots represent the ten joints to be estimated while the yellow dots represent the unchosen joints. Six hand movements are introduced and shown in (b).

correlation between the predicted and actual movements, denoting enhanced performance in motion estimation.

2) **Normalized Root Mean Square Error (NRMSE)**: NRMSE, a derivative of Root Mean Square Error (RMSE), quantifies the deviation between predicted and observed values. It is normalized for joint angles as follows:

$$NRMSE = \frac{RMSE}{\theta_{max} - \theta_{min}} \quad (10)$$

where θ_{max} and θ_{min} represent the maximum and minimum true joint angle values, facilitating comparison across different joints.

3) **Unbiased Standard Deviation (σ)**: The unbiased standard deviation (σ) measures the dispersion of data, assessing the stability of our method. The σ for 10 joints per subject is calculated, with smaller values indicating reduced dispersion and thus higher estimation consistency.

4) **Lifelong Learning Evaluation**: For evaluating lifelong learning performance, we adopt criteria from gradient episodic memory [18]:

Average Accuracy on Tasks (ACCT): This metric reflects the overall proficiency of the model across all tasks, indicating its ability to preserve prior knowledge while adapting to new tasks.

$$ACCT = \frac{1}{T} \sum_{i=1}^T R_{T,i} \quad (11)$$

where $R_{i,j}$ means the performance of model trained after previous i tasks performed on the j^{th} task. T means the total number of tasks, and the same applies throughout.

Average Accuracy on Subjects (ACCS): Focused on the final model, ACCS evaluates the average accuracy across all subjects, indicating the model's effectiveness in retaining knowledge and adapting across subjects.

$$ACCS = \frac{1}{N} \sum_{i=1}^N CC(Y_i, \hat{Y}_i) \quad (12)$$

where CC represents the calculation of PCC, Y_i denotes the true movement of the i^{th} subject and \hat{Y}_i denotes the predicted movement of the i^{th} subject.

Backward Transfer (BWT): This measures the effect of learning new tasks on the performance of previous tasks, revealing the capability to maintain performance on earlier tasks.

$$BWT = \frac{1}{T-1} \sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \quad (13)$$

Forward Transfer (FWT): FWT assesses the impact of prior learning on new tasks, indicating the ability to utilize previous knowledge for new tasks.

$$FWT = \frac{1}{T-1} \sum_{i=2}^T R_{i-1,i} - \bar{b}_i \quad (14)$$

The Wilcoxon signed-rank test, adjusted by the Bonferroni correction, is used to evaluate the statistical significance of our method.

C. Baseline Methods

In this study, our method is compared against two transfer learning approaches (fine-tuning and domain adaptation) and two lifelong learning techniques (EWC and LwF).

Fine-tuning adapts pre-trained models to specific tasks, refining parameters with new data. Domain adaptation transfers knowledge between different domains, assessing the ability of our model to mitigate catastrophic forgetting.

EWC and LwF are lifelong learning methods focusing on knowledge preservation from previous tasks. EWC regularizes model parameters based on their task significance, while LwF utilizes knowledge distillation for transitioning knowledge between tasks. These methods evaluate the efficacy of our strategy in a lifelong learning context.

D. Experimental Results

The effectiveness of our method in continuous hand movement estimation tasks is rigorously evaluated against previous cross-subject approaches. All models are developed using the PyTorch framework [43] and trained on an NVIDIA GeForce RTX 3090 GPU with adam optimizer. The training process spans 400 epochs, with 200 epochs dedicated to each transfer phase. Our experiments are conducted under two distinct scenarios: cross-subject training and cross-subject lifelong learning.

In the cross-subject training scenario, models are simultaneously trained and validated using data from all 40 subjects. For cross-subject lifelong learning, models undergo sequential

TABLE II
PARAMETER STUDY OF μ W.R.T. CC AND NRMSE ON ALL 40 SUBJECTS

μ	2^5	2^{10}	2^{15}	2^{20}	2^{25}
CC	0.7191	0.7820	0.8422	0.8552	0.8566
NRMSE	0.1088	0.0968	0.0817	0.0782	0.0784

TABLE III
AVERAGE PERFORMANCE OF DIFFERENT MODELS ON 10 JOINTS AND 6 MOVEMENTS OF 40 DIFFERENT SUBJECTS IN CROSS-SUBJECT SETTING

Model	CC	NRMSE	σ_c	σ_n
BERT[5]	0.8052	0.0925	0.095	0.023
sBERT[5]	0.8300	0.0872	0.090	0.027
BERT-GHM[5]	0.7906	0.0970	0.106	0.029
sBERT-GHM[5]	0.7878	0.0946	0.108	0.039
BERT-OHEM[5]	0.7768	0.0984	0.108	0.025
sBERT-OHEM[5]	0.7975	0.0934	0.096	0.032
TCN	0.8193	0.0874	0.089	0.025
CSLN(-Decoders)	0.8533	0.0784	0.075	0.023
CSLN	0.8552	0.0782	0.072	0.019

training across tasks segmented by subjects. To assess performance, we calculate metrics such as NRMSE and PCC, along with monitoring the average training time per epoch and the total convergence time for each training session.

Parameter selection was meticulously conducted after extensive trials with various configurations. We standardized all models to train with a learning rate of 0.0001, which is halved post-200 epochs. The μ parameter in μ -law normalization is fixed at 2^{20} , as determined from a parameter range study spanning 2^5 to 2^{25} . The results of this study are detailed in Table II. Additionally, the λ parameter is set at 1×10^4 , optimizing the balance between different loss components.

1) *Cross-Subject Settings*: To demonstrate the efficacy of our method in a cross-subject context, we conducted comparative training with state-of-the-art cross-subject models [5] and our CSLN. This training encompassed all 40 subjects, and the results are detailed in Table III.

Our method showcases the superior performance, achieving average Pearson Correlation Coefficient (CC) and Normalized Root Mean Squared Error (NRMSE) values of 0.8552 ± 0.08 and 0.0782 ± 0.03 , respectively, for individual subjects. These results significantly outperform all BERT-based methods, including standard BERT (0.8051 ± 0.08 , $p < 0.005$; 0.09 ± 0.03 , $p < 0.005$), sBERT (0.8300 ± 0.09 , $p < 0.005$; 0.09 ± 0.03 , $p < 0.005$), BERT-GHM (0.7908 ± 0.14 , $p < 0.005$; 0.10 ± 0.03), sBERT-GHM (0.7878 ± 0.09 , $p < 0.005$; 0.09 ± 0.03 , $p < 0.005$), BERT-OHEM (0.7768 ± 0.10 , $p < 0.005$; 0.10 ± 0.03), and sBERT-OHEM (0.7975 ± 0.11 , $p < 0.005$; 0.09 ± 0.03 , $p < 0.005$), as confirmed by the Wilcoxon signed-rank test.

An interesting observation is that the training convergence time increased with the number of subjects included. However, it is noteworthy that in the lifelong learning setup, the convergence time per subject was, on average, approximately 120 seconds shorter compared to direct task training. This indicates enhanced efficiency in the lifelong learning scenario.

2) *Lifelong Learning Settings*: In the lifelong learning experiments, a random shuffle is performed on the 40 subjects from the Ninapro DB2 dataset. These subjects are then randomly divided into 5 tasks, forming various task sequences. To mitigate potential biases, this process is repeated, creating

TABLE IV
PERFORMANCE OF EVERY TASK SEQUENCE ON 10 JOINTS AND 6 MOVEMENTS OF 40 DIFFERENT SUBJECTS IN LIFELONG LEARNING SETTING WITH DIFFERENT LEARNING STRATEGIES

Learning Strategy	Sequence	Metrics			
		ACCS \uparrow	ACCT \uparrow	BWT \uparrow	FWT \uparrow
FineTuning	S_1	0.6026	0.5995	-0.3393	0.5427
	S_2	0.5545	0.5685	-0.3791	0.6071
	S_3	0.5991	0.5922	-0.3431	0.5604
	S_4	0.5881	0.5677	-0.3820	0.4152
	S_5	0.5588	0.6067	-0.3368	0.4570
	Avg	0.5806	0.5869	-0.3561	0.5165
Domain Adaptation	S_1	0.5903	0.5878	-0.3556	0.5330
	S_2	0.5278	0.5361	-0.4240	0.6321
	S_3	0.3741	0.3784	-0.3565	0.5760
	S_4	0.5717	0.5573	-0.3955	0.4269
	S_5	0.5252	0.5776	-0.3736	0.4690
	Avg	0.5178	0.5274	-0.3810	0.5274
LwF[21]	S_1	0.6849	0.6834	-0.1717	0.6005
	S_2	0.6622	0.6724	-0.1977	0.6661
	S_3	0.6925	0.6861	-0.1751	0.6469
	S_4	0.6800	0.6750	-0.1890	0.4804
	S_5	0.6492	0.6833	-0.1931	0.5039
	Avg	0.6738	0.6800	-0.1853	0.5796
EWC[20]	S_1	0.6313	0.6293	-0.2822	0.5530
	S_2	0.5681	0.5751	-0.3420	0.6297
	S_3	0.6169	0.6118	-0.2985	0.5841
	S_4	0.5946	0.5899	-0.3282	0.4290
	S_5	0.5782	0.6212	-0.3016	0.4826
	Avg	0.5978	0.6055	-0.3105	0.5357
Ours	S_1	0.7613	0.7640	-0.1085	0.5156
	S_2	0.7708	0.7741	-0.0952	0.6077
	S_3	0.7664	0.7667	-0.1035	0.5600
	S_4	0.7821	0.7716	-0.1009	0.4300
	S_5	0.7710	0.7847	-0.0914	0.4358
	Avg	0.7703	0.7722	-0.0999	0.5098

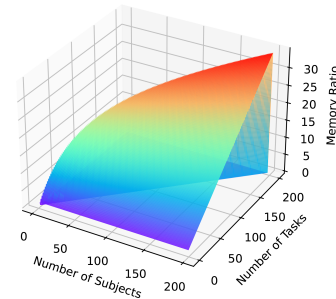


Fig. 4. The memory ratio, which is defined as the ratio of memory usage between the model after direct training or transferring and the model after the lifelong learning procedure.

5 task sequences as documented in Table I. These sequences are referred to as S_i (where $i \in 1, \dots, 5$), with their detailed division provided in Table I. The evaluation criteria, based on the correlation coefficient (CC), are presented in Table IV.

Furthermore, significant improvements are observed in the average performance in terms of ACCT and ACCS, compared to other approaches. The consistent results in FWT across various methods underscore the comparable efficacy of the proposed strategy.

A detailed analysis reveals that lifelong learning methods excel in BWT, surpassing transfer learning methods and affirming their superiority in retaining knowledge. This demonstrates their effectiveness in preserving previously acquired knowledge while adapting to new tasks. Specifically, our lifelong learning approach shows exemplary performance in BWT, ACCT, and ACCS, underscoring its robustness in knowledge retention. However, a lower performance in

TABLE V

DETAILED CC RESULT OF EACH TASK OF TASK SEQUENCE S_1 ON 10 JOINTS AND 6 MOVEMENTS OF 40 DIFFERENT SUBJECTS IN LIFELONG LEARNING SETTING WITH DIFFERENT LEARNING STRATEGIES

After Training on	Strategies	The Model Evaluated on					ACCS
		T_1	T_2	T_3	T_4	T_5	
Random Initialized(b)	–	-0.4244	0.2968	-0.4143	-0.0522	0.0284	–
T_1	–	0.8700	0.5019	0.5272	0.5072	0.4868	0.8700
T_2	Fine-Tuning	0.4854	0.8539	0.5076	0.4994	0.5275	0.7238
	Domain Adaptation	0.4874	0.8652	0.5158	0.4788	0.5566	0.7319
	LwF	0.7074	0.7506	0.5774	0.5572	0.5697	0.7354
	EWC	0.5242	0.8481	0.5376	0.5175	0.5769	0.7338
	Ours	0.8228	0.8431	0.4912	0.4264	0.4504	0.8359
T_3	Fine-Tuning	0.5404	0.6021	0.8795	0.5054	0.5103	0.6584
	Domain Adaptation	0.5644	0.5759	0.8808	0.4748	0.5196	0.6524
	LwF	0.7005	0.6763	0.8156	0.5806	0.5859	0.7190
	EWC	0.5659	0.5905	0.8652	0.5277	0.5436	0.6557
	Ours	0.8407	0.7330	0.8399	0.4572	0.4782	0.7890
T_4	Fine-Tuning	0.4725	0.4984	0.5376	0.8818	0.5146	0.6650
	Domain Adaptation	0.4696	0.4752	0.5130	0.8800	0.4982	0.5951
	LwF	0.6305	0.6313	0.6978	0.8341	0.6006	0.7007
	EWC	0.4824	0.5193	0.5838	0.8514	0.5035	0.6179
	Ours	0.8159	0.7117	0.6387	0.8572	0.4782	0.7584
T_5	Fine-Tuning	0.5024	0.5460	0.5293	0.5504	0.8693	0.6026
	Domain Adaptation	0.4871	0.5360	0.5287	0.5219	0.8653	0.5903
	LwF	0.6117	0.6404	0.6576	0.6738	0.8337	0.6849
	EWC	0.5534	0.5914	0.5822	0.5790	0.8404	0.6313
	Ours	0.8241	0.7055	0.6735	0.7733	0.8434	0.7613

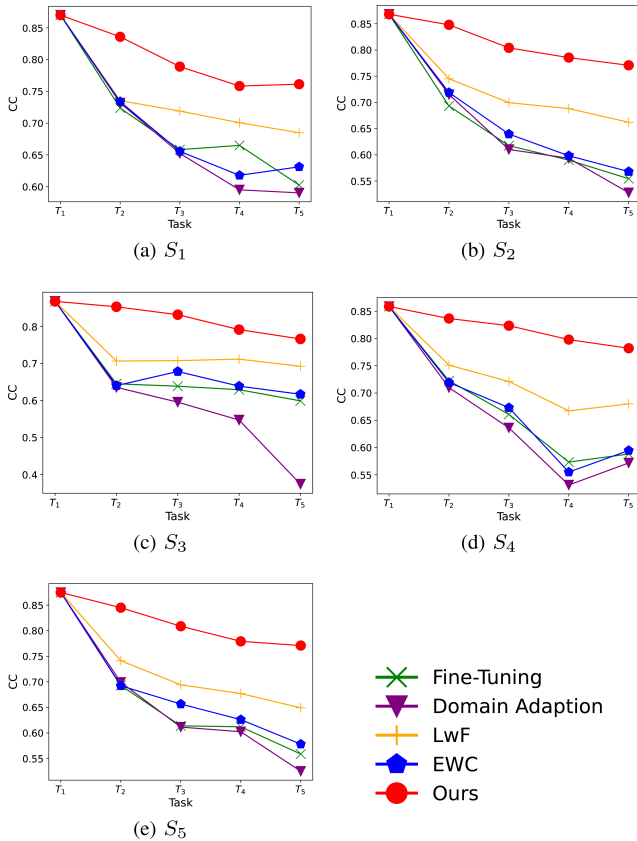


Fig. 5. The variation of ACCS after each training procedure using different strategies on different task sequences.

the FWT criterion indicates a potential avenue for further enhancement.

The experimental results reveal that the arrangement and order of subjects minimally impact the performance of our method. Similar levels of performance are observed across all generated task sequences in various evaluation criteria, indicating that the specific arrangement of subjects within task sequences does not significantly influence the overall efficacy. This finding implies that flexibility in selecting subject

division and order can be tailored to practical requirements without adversely affecting the effectiveness of the lifelong learning approach. To demonstrate the impact of knowledge retention on overall performance, Fig. 5 illustrates the variation in ACCS following each training session using different strategies.

As depicted in Figure 6, the knowledge retention performance of our method is showcased on two channels of subject s8, corresponding to task T_1 in task sequence S_1 from the Ninapro DB2 dataset. The figure highlights three distinct periods for each strategy: after training on T_1 , after training on T_3 , and after completing the final task, T_5 .

While our method effectively mitigates catastrophic forgetting, it cannot completely eliminate it. Each new task introduces a heightened risk of forgetting the old tasks, inevitably leading to a decline in average performance across all tasks, especially as the number of tasks in a sequence increases. Here, we present an extreme case involving 40 subjects divided into 40 tasks, resulting in an accuracy shrinkage to 0.70. However, even in this scenario, our method still outperforms other methods on 5 tasks, underscoring its superiority.

Furthermore, the lifelong learning strategy exhibits accelerated convergence compared to training models from scratch for each task. Utilizing knowledge from previous tasks facilitates more efficient learning on subsequent tasks, thereby reducing the convergence time. This efficiency gain enhances training speed, potentially expediting model deployment in practical scenarios.

Regarding memory usage, the lifelong learning model consistently outperforms models that are directly trained or fine-tuned. The CSLN model maintains a fixed memory footprint based on the number of subjects, whereas the memory requirements of directly trained models escalate with the number of tasks. In scenarios with 40 subjects divided into 40 tasks, the memory consumption of direct training or fine-tuning models can be nearly 20 times greater than that of the lifelong learning approach. However, achieving an optimal scenario of one task for all 40 subjects remains a challenge in real-world contexts. As visualized in Figure 4, the memory ratio—defined as the ratio of memory usage between models post-direct training or transfer and post-lifelong learning—consistently favors lifelong learning models, with a ratio greater than or equal to 1. This consistent efficiency advantage highlights the value of lifelong learning in optimizing memory usage.

V. DISCUSSION

The CSLN model, as proposed in this study, demonstrates remarkable efficacy in estimating finger joint movements from sEMG signals for the environments that need a singular, adaptable model. It exhibits superior performance in both cross-subject and lifelong learning scenarios. Moreover, the adaptability to various temporal scales is implied by the utilization of Root Mean Square (RMS) features extracted with a sliding window, allowing it to accommodate sEMG data at different temporal resolutions. Additionally, its cross-subject training capability suggests the ability to generalize across individuals, including variations in the temporal patterns of their sEMG signals, further highlighting its practical

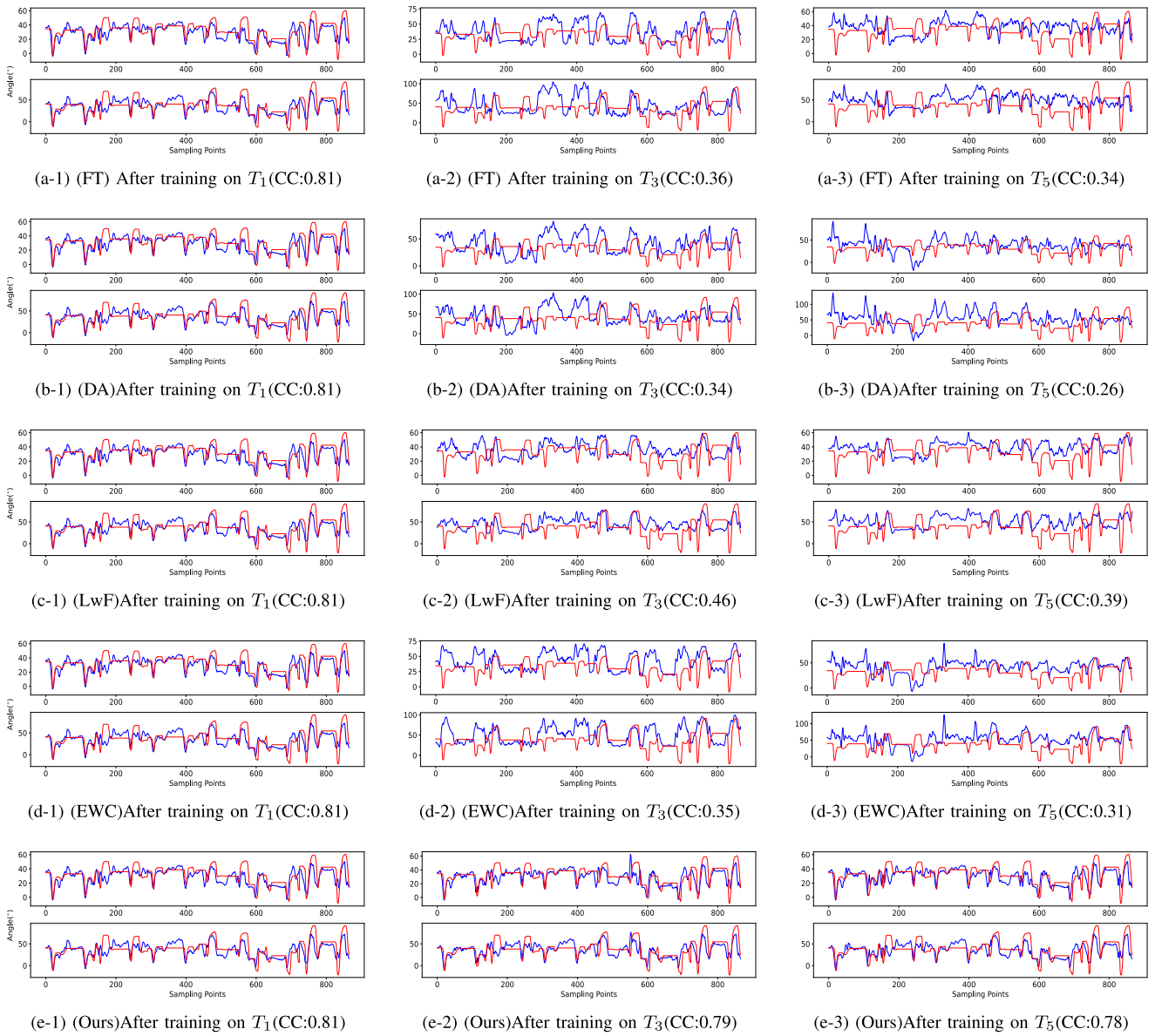


Fig. 6. Performance Variation of Model on Subject s8 from T_1 of S_1 . The figure displays the estimation results of subject s8, comparing method performance across three periods: after training on T_1 , T_3 , and the final task T_5 . “FT” and “DA” denote fine-tuning and domain adaptation techniques, respectively. Red curves represent ground truth, while blue curves depict the estimation. All the results are smoothed by the AvgSmooth method mentioned in [5].

relevance in the field. Utilizing evaluation metrics such as CC and NRMSE, the model achieves a best CC of 0.8552 and an NRMSE of 0.0782, underscoring its advancement over previous cross-subject models, particularly for subjects with intact hands. Such results solidify the position of our method as state-of-the-art in cross-subject settings, highlighting its practical relevance in the field.

In addition, the novel lifelong learning method integrated into CSLN demonstrates remarkable efficiency in training. This approach surpasses traditional model training from random initialization, showcasing enhanced knowledge retention and faster adaptation to new tasks. With the highest BWT of about -0.10, the model leverages adapters and a memory module for more rapid convergence and reduced training duration. The model’s expansion, facilitated by lightweight adapters, not only augments its practicality and scalability but also achieves optimal memory utilization, requiring approximately only

one-twentieth of the parameters compared to other methods. However, the underutilization of previously learned knowledge may account for the lower performance in the FWT.

While the CSLN model presents a versatile solution for cross-subject sEMG applications, it is also crucial to acknowledge certain limitations. The model’s assumption of distinct subjects within each task sequence may affect its generalization to scenarios with repeated subjects, potentially impacting real-world applicability. There is also room for improvement in inference efficiency, which is a vital aspect for practical deployment.

VI. CONCLUSION

This paper introduces the CSLN, a novel method for accurate continuous hand gesture estimation. Employing a lifelong learning strategy, CSLN adeptly navigates the complexities

of hand movement dynamics, effectively addressing cross-subject task challenges and significantly reducing catastrophic forgetting. Our method outperforms existing approaches in terms of accuracy and knowledge retention, making it a promising solution for practical applications such as in factory settings and smart home environments. These scenarios demand versatile methods that adapt seamlessly across various subjects and diverse spatial-temporal contexts. The application of CSLN to wider domains and an array of hand motion tasks promises to enhance its practical utility further. Future research directions include expanding the capabilities to encompass a broader spectrum of hand gesture dynamics and exploring its potential in other relevant fields. The aim is to contribute substantially to the advancement of hand motion estimation systems, fostering the development of robust and adaptable solutions.

REFERENCES

- [1] M. R. Ahsan et al., “Emg signal classification for human computer interaction: A review,” *Eur. J. Sci. Res.*, vol. 33, no. 3, pp. 480–501, 2009.
- [2] P. D. Lawrence and W.-C. Lin, “Statistical decision making in the real-time control of an arm aid for the disabled,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 1, pp. 35–42, Mar. 1972.
- [3] M. Långkvist, L. Karlsson, and A. Loutfi, “A review of unsupervised feature learning and deep learning for time-series modeling,” *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.
- [4] W. Guo et al., “Long exposure convolutional memory network for accurate estimation of finger kinematics from surface electromyographic signals,” *J. Neural Eng.*, vol. 18, no. 2, Apr. 2021, Art. no. 026027.
- [5] C. Lin, X. Chen, W. Guo, N. Jiang, D. Farina, and J. Su, “A BERT based method for continuous estimation of cross-subject hand kinematics from surface electromyographic signals,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 87–96, 2023.
- [6] Q. Zhang, R. Liu, W. Chen, and C. Xiong, “Simultaneous and continuous estimation of shoulder and elbow kinematics from surface EMG signals,” *Frontiers Neurosci.*, vol. 11, p. 280, May 2017.
- [7] F. Quivira, T. Koike-Akino, Y. Wang, and D. Erdogmus, “Translating sEMG signals to continuous hand poses using recurrent neural networks,” in *Proc. IEEE EMBS Int. Conf. Biomed. Health Informat. (BHI)*, Mar. 2018, pp. 166–169.
- [8] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 802–810.
- [9] 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.
- [10] J. Fan et al., “Improving sEMG-based motion intention recognition for upper-limb amputees using transfer learning,” *Neural Comput. Appl.*, vol. 35, no. 22, pp. 16101–16111, Aug. 2023.
- [11] Y. Long, Y. Geng, C. Dai, and G. Li, “A transfer learning based cross-subject generic model for continuous estimation of finger joint angles from a new user,” *IEEE J. Biomed. Health Informat.*, vol. 27, no. 4, pp. 1914–1925, Apr. 2023.
- [12] U. Côté-Allard et al., “Deep learning for electromyographic hand gesture signal classification using transfer learning,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.
- [13] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Lavolette, and B. Gosselin, “Transfer learning for sEMG hand gestures recognition using convolutional neural networks,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 1663–1668.
- [14] Y. D. Kwon, J. Chauhan, A. Kumar, P. H. Hkust, and C. Mascolo, “Exploring system performance of continual learning for mobile and embedded sensing applications,” in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Dec. 2021, pp. 319–332.
- [15] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, “Embracing change: Continual learning in deep neural networks,” *Trends Cogn. Sci.*, vol. 24, no. 12, pp. 1028–1040, Dec. 2020.
- [16] H. Chen, Y. Wang, and Q. Hu, “Multi-granularity regularized rebalancing for class incremental learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 7263–7277, Jul. 2022.
- [17] Y. Yang et al., “Learning adaptive embedding considering incremental class,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2736–2749, Mar. 2023.
- [18] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6467–6476.
- [19] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, “Efficient lifelong learning with A-GEM,” in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, May 2019.
- [20] K. James et al., “Overcoming catastrophic forgetting in neural networks,” *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.
- [21] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2017.
- [22] M. Boschini et al., “Transfer without forgetting,” in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2022, pp. 692–709.
- [23] A. V. Terekhov, G. Montone, and J. K. O’Regan, “Knowledge transfer in deep block-modular neural networks,” in *Proc. 4th Int. Conf. Biomimetic Biohybrid Syst.*, Barcelona, Spain. Cham, Switzerland: Springer, 2015, pp. 268–279.
- [24] A. A. Rusu et al., “Progressive neural networks,” 2016, *arXiv:1606.04671*.
- [25] H. Mostafa and X. Wang, “Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization,” in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 4646–4655.
- [26] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4548–4557.
- [27] A. Robins, “Catastrophic forgetting, rehearsal and Pseudorehearsal,” *Connect. Sci.*, vol. 7, no. 2, pp. 123–146, 1995.
- [28] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, “Experience replay for continual learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 348–358.
- [29] C. de Masson D’Autume, S. Ruder, L. Kong, and D. Yogatama, “Episodic memory in lifelong language learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 13122–13131.
- [30] P. Sprechmann et al., “Memory-based parameter adaptation,” in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018.
- [31] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 2990–2999.
- [32] A. Rannen, R. Aljundi, M. B. Blaschko, and T. Tuytelaars, “Encoder based lifelong learning,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1320–1328.
- [33] D. Rao, F. Visin, A. Rusu, R. Pascanu, Y. W. Teh, and R. Hadsell, “Continual unsupervised representation learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 7645–7655.
- [34] S. Flennerhag, A. A. Rusu, R. Pascanu, F. Visin, H. Yin, and R. Hadsell, “Meta-learning with warped gradient descent,” in *Proc. 8th Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, 2020.
- [35] S. Beaulieu et al., “Learning to continually learn,” in *Proc. ECAI*, in *Frontiers in Artificial Intelligence and Applications*, vol. 325, 2020, pp. 992–1001.
- [36] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, “Xceptiontime: Independent time-window xceptiontime architecture for hand gesture classification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2020, pp. 1304–1308.
- [37] *Pulse Code Modulation (PCM) of Voice Frequencies*, ITU, Geneva, Switzerland, 1988.
- [38] N. Houlsby et al., “Parameter-efficient transfer learning for NLP,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.
- [39] S. Bai, J. Zico Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, *arXiv:1803.01271*.
- [40] G. Kim, C. Xiao, T. Konishi, Z. Ke, and B. Liu, “A theoretical study on solving continual learning,” 2022, *arXiv:2211.02633*.
- [41] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, “Learning multiple visual domains with residual adapters,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–13.
- [42] M. Atzori et al., “Electromyography data for non-invasive naturally-controlled robotic hand prostheses,” *Sci. Data*, vol. 1, no. 1, pp. 1–13, Dec. 2014.
- [43] N. Ketkar and J. Moolayil, “Introduction to PyTorch,” in *Deep Learning With Python*. Berkeley, CA, USA: Apress, 2021, pp. 27–91.