Article:
Zhang, J., Ruan, Z., Li, Q. et al. (2023) Toward Robust and Efficient Musculoskeletal Modeling Using Distributed Physics-Informed Deep Learning. IEEE Transactions on Instrumentation and Measurement, 72. 2530511. ISSN: 0018-9456

https://doi.org/10.1109/tim.2023.3325522

Towards Robust and Efficient Musculoskeletal Modelling Using Distributed Physics-informed Deep Learning

Jie Zhang, *Member, IEEE*, Ziling Ruan, Qing Li, Zhi-Qiang Zhang, *Member, IEEE*

***Abstract*—This paper develops a novel distributed framework based on physics-informed deep learning for robust and efficient musculoskeletal modelling in nonstationary scenarios, which could simultaneously strengthen the robustness and generalization, and reduce the time cost of model training. Without loss of generality, we utilize surface electromyogram (sEMG)-based muscle forces and joint angle prediction as an example to demonstrate the proposed distributed framework. Specifically, the whole collected sEMG data are first divided into several sub-domains, and then corresponding number of physics-informed deep learning-based local models is built using these grouped data. Finally, all the local models are integrated into a global model to obtain the final predictions. Moreover, weights inversely proportional to the training errors of local models are added to the corresponding local models to reduce and control negative effects of unknown factors. Different from existing distributed modelling methods, the proposed distributed framework embeds the prior physics knowledge, i.e., the equation of motion, into local models to regularise loss functions of deep neural networks, it thus could overcome limitations of the conventional data-driven and physics-based musculoskeletal models while preserving their advantages. The local-global distributed modelling mechanism could locally achieve better representation for sub-domains while preserving global performance, and reduce the computational cost and memory requirements. Additionally, the embedded prior physics knowledge enables local models to reflect physical or physiological mechanisms during the training process, which could alleviate overfitting problem and reduce the need of the number of training data, and thus the global model is more robust and better generalizes to the unseen data. Comprehensive experiments on six healthy subjects demonstrate the feasibility and effectiveness of the proposed distributed framework.**

***Index Terms*—Musculoskeletal modelling, muscle forces and joint angle prediction, physics-informed deep learning, local-global distributed modelling.**

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S019219/1, and in part by the European Union (EU) Marie Curie Individual Fellowship under Grant 101023097.

Jie Zhang and Zhi-Qiang Zhang are with the School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, U.K. (e-mails: eenjz@leeds.ac.uk, z.zhang3@leeds.ac.uk).

Ziling Ruan is with the Key Laboratory of Smart Human Computer Interaction and Wearable Technology of Shaanxi Province, School of Computer Science and Technology, Xidian University, Xi'an, 710071, China; and also with the School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, U.K. (e-mail: 21031211775@stu.xidian.edu.cn).

Qing Li is with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China (e-mail: liqing@ies.ustb.edu.cn).

I. Introduction

HUMAN movement is actually a complex interaction of neuromusculoskeletal system, but the dynamical properties of biological components, including muscle and tendon, could be altered by aging, injury or disease, which may change how neural commands are converted to muscle forces and joint torques [1]–[5]. Therefore, understanding how the neuromusculoskeletal system learns and adapts to the physiological modification is crucial for describing the dynamics of human movements [6], [7]. Musculoskeletal modelling has been commonly utilized to quantify the neuromuscular activity, and could provide important insights into the mechanics and control of human movements, which has great benefits of patients with motor dysfunction to recover control ability of the motor cortex and strengthen the athletic ability of athletes [8]–[10]. However, most of existing physics-based musculoskeletal modelling approaches are time-consuming, limiting the large-scale implementation in various clinical applications [11]–[14].

In the past years, data-driven musculoskeletal models have been widely developed based on machine/deep learning techniques [15], [16]. Specifically, Hajian et al. [17] implemented a two streams of convolutional neural network (CNN) to extract discriminative features from the raw electromyogram (EMG) signals using different scales, and predicted the generated motions during elbow flexion and extension. Dao et al. [18] first designed a recurrent deep neural network for skeletal muscle force prediction, and then developed a modified transfer learning strategy to enhance the performance. Yang et al. [19] proposed a modified CNN to decode wrist movements of multiple degrees of freedom (DoF) from the raw EMG singals. Kim et al. [20] designed a modified deep transfer learning approach for the hand movement estimation, in which CNNs were first pre-trained utilizing several subjects' EMG data, and then fine-tuned for the targeted subject through the signal-trial analysis. Compared with physics-based modelling methods, the inference of machine/deep learning-based data-driven musculoskeletal modelling approaches are much faster once the neural networks are well-trained [21], [22].

Although data-driven musculoskeletal modelling methods have made promising achievements, there are still some challenging issues should be addressed when we want to implement such data-driven musculoskeletal models in real-world application scenarios. Firstly, machine/deep learning are the "black-box" tools to build the nonlinear mapping

relationship between the input and output, and the training usually does not associate with the mechanisms underlying the observed variables [23], [24]. It means that existing data-driven musculoskeletal models are without interpretable neuromechanical processes [25], [26]. Secondly, most of existing data-driven musculoskeletal modelling methods mainly focus on prediction accuracy improvement, but ignore the robustness. When a high-accuracy data-driven musculoskeletal model is implemented in real-world application scenarios, unexpected results may still be obtained due to various unknown factors, such as shift of the electrode and muscle fatigue [27], leading to the created model unstable in robustness [28], [29]. Finally, the training of deep neural networks with complicated architectures is very time-consuming especially when the amount of the available training data is large or the dimension of the training data is high, the large training cost associated with deep neural networks thus cannot be ignored in practical scenarios [30]–[32].

To tackle the above challenges of existing musculoskeletal modelling methods, a novel distributed framework based on physics-informed deep learning is proposed for musculoskeletal modelling in non-stationary scenarios by simultaneously strengthening the robustness and generalization, and reducing the time cost in model training. Without loss of generality, we utilize surface EMG (sEMG)-based muscle forces and joint angle prediction as an example to demonstrate the proposed distributed framework. Specifically, the whole sEMG data are first divided into several sub-domains, and then corresponding number of local models is built using these grouped data with physics-informed deep learning. Different from existing deep learning-based musculoskeletal modelling methods, the equation of motion is considered as the prior physics knowledge and embedded into local models to regularise the loss functions of deep neural networks, it thus could overcome limitations of existing data-driven and physics-based musculoskeletal modelling methods while preserving their advantages. Finally, all the local models are integrated into a global model to obtain the final predictions, such distributed modelling mechanism could accelerate the training of the deep learning-based model. Moreover, weights inversely proportional to the training errors of local models are added to the corresponding local models to reduce and control negative effects of unknown factors. In summary, the main advantages of the proposed distributed framework are multi-fold:

- **Representation capability.** The proposed distributed framework develops a set of individual local models with different characteristics complied to the prior physics knowledge to represent the grouped sEMG data separately, such physics-informed deep learning-based models enable intermediate functional relationships to reflect the mechanisms underlying the observed variables. Additionally, local models could flexibly select appropriate architectures and hyperparameters (such as depth and width of networks, and activation function types, etc.), and even implement different types of networks depending on the statistical characteristics of sub-domains. Therefore, it makes the conventional "black-box" modelling process more interpretable, and could locally achieve better representation for sub-domains while preserve global performance.
- **Efficient training.** The partial independence of local models in the divided sub-domains enables a set of parallel computations, thereby reducing the computational cost and memory requirements. Additionally, smaller number of training data in each sub-domain usually requires simpler network architectures and less hyperparameters to be adjusted, which can avoid the overfitting problem, and thus its training process is more efficient.
- **Robustness and generalization.** Weighting strategy enables local models with smaller training errors to dominate the global performance, even some grouped data are seriously contaminated by unknown factors, the created global model still could achieve satisfactory performance. Furthermore, the embedded prior physics knowledge makes local models reflect physical or physiological mechanisms during the training process, which also could alleviate the overfitting problem and reduce the need of the number of training data, making the global model be with better robustness and generalization.

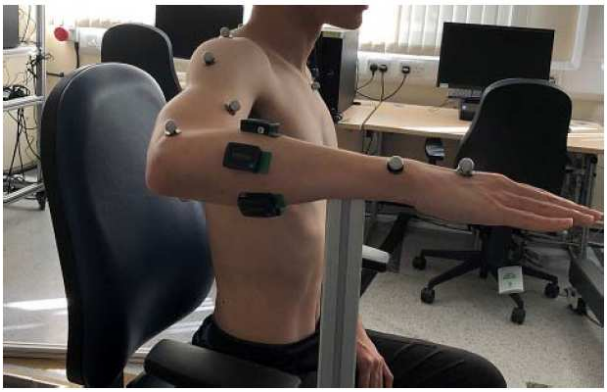


Fig. 1. Electrodes and markers placement in the data collection process.

The remaining of the paper is organised as follows: Details of the data collection and processing, data partition, dataset construction, the proposed distributed framework and error analysis are given in Section II. Experimental results and corresponding analysis are shown in Section III. Advantages and limitations of the proposed distributed framework, and future works are discussed in Section IV. Finally, conclusions are summarized in Section V.

II. Methodology

In this section, the data collection and processing, data partition and dataset construction for domain decomposition are first shown, the proposed distributed framework is then detailed, including the main framework, architecture, training and loss function of local models, aggregation of local models for global modelling, and error analysis.

A. Data Collection and Processing

The experimental data were collected from six able-bodied volunteers, approved by the MaPS and Engineering Joint

Faculty Research Ethics Committee of University of Leeds (MEEC 18-002). During the data collection, all the volunteers were asked to keep a fully straight torso with the 90° abducted shoulder and 90° flexed elbow joint supporting by the vertical bar (see Fig. 1), and continuous wrist flexion and extension motions were captured by the VICON system. Additionally, joint motions could be computed by the upper limb model with reflective markers (the sampling rate is 250 Hz), and sEMG signals were measured using Avanti sensors (the sampling rate is 2000 Hz) from five wrist muscles, including the extensor carpi radialis longus (ECRL), flexor carpi radialis (FCR), extensor carpi radialis brevis (ECRB), flexor carpi ulnaris (FCU), and extensor carpi ulnaris (ECU). Each volunteer was asked to perform five repetitive trials, and a short break was done between trials to avoid negative effects of the muscle fatigue on the data quality. Both the motion data and EMG measurements were synchronised and resampled at 1000 Hz. After that, EMG measurements were processed with band-pass filtered, fully rectified, and low-pass filtered, and then they were normalized concerning the maximum voluntary contraction (MVC) recorded before experiments, resulting in the enveloped EMG data. MVC represents the peak EMG activity observed during a task and provides an estimation of the theoretical maximum amplitude of the muscle activation during a contraction. The normalized EMG signal is presented as a percentage of the maximum activation level [33]. The marker's data were utilized to compute the wrist kinematics through the inverse kinematic (IK) tool based on the upper limb extremity model [34]. The joint torque and wrist muscle forces were obtained from the inverse dynamic (ID) and computed muscle control (CMC) tools. Specifically, we utilized the ID tool to calculate the joint torques based on the input joint angles, and the CMC tool to generate a set of muscle excitation levels to control muscle forces, enabling them to drive the desired kinematic trajectory [35], [36].

B. Data Partition and Dataset Construction

The data collected from six volunteers were partitioned into several sub-domains for the distributed modelling purpose. Specifically, among the collected data, for every ten samples, we selected the first seven samples as one training candidate group to construct the training dataset, and the remaining three samples as one testing candidate group to construct the testing dataset. After that, training dataset was constructed by training candidate groups, while the testing dataset consisted of testing candidate groups. The constructed training dataset was then divided into several sub-domains, the number of sub-domains is equal to the number of local models, and the number of data in each of sub-domains should be the same or at least not significantly different. It should be noted that the data from each muscle were included in each sub-domain, and the range of sub-domains is between 3 and 5 in this study.

C. Main Framework of Distributed Physics-informed Deep Learning for Musculoskeletal Modelling

As shown in Fig. 2, we utilize CNN as the baseline network in local models to demonstrate the proposed distributed framework. Specifically, all the collected sEMG data are first divided into Q sub-domains, and then Q local models are created using these grouped data in a parallel manner. Finally, all the local models are integrated into the global model to obtain the final predictions. During the aggregation of local models, weighting strategy is considered to reduce and control negative effects of unknown factors imposed on local models. In this way, local models with smaller training errors could dominate the global performance, making the created global model more robust.

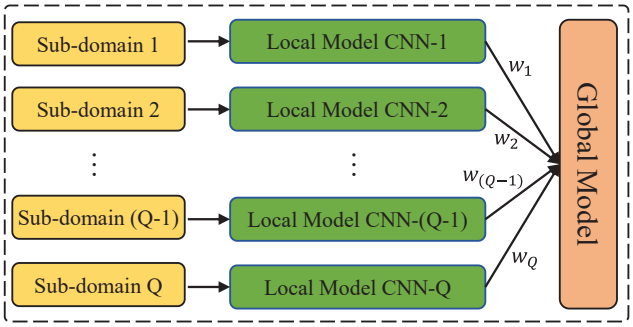


Fig. 2. CNN is used as the baseline network of local models to demonstrate the proposed distributed framework, and Q (q=1,2,...,Q) denotes the number of local models. Weighting strategy is implemented to reduce and control the negative effects of unknown factors imposed on local models.

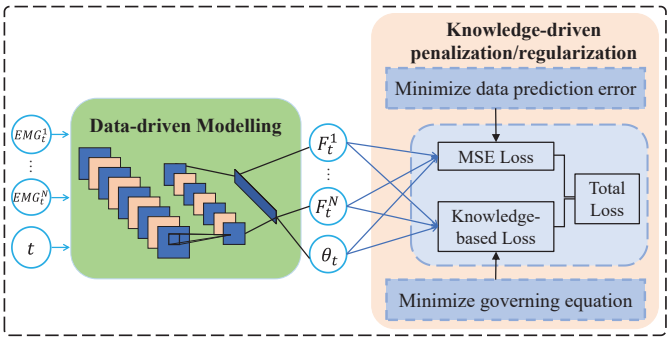


Fig. 3. Illustration of the physics-informed deep learning-based local model. Inputs are sEMG measurements and the corresponding time steps, while outputs are muscle forces F_t^n and joint angles θ_t ($n = 1, \ldots, N, t = 1, \ldots, T$).

D. Training of Local Models

As demonstrated in Fig. 3, training local models involves two phases, including the data-driven modelling phase and knowledge-driven regularization phase. Specifically, the collected sEMG data and corresponding time steps are used as the inputs of local models, and the predictions of muscle forces and joint angle are then obtained with the extracted high-level features. Different from existing methods, the predicted outputs of CNNs are also guided by the prior physics knowledge in the loss function. Therefore, the modified loss function is to simultaneously minimize both the data prediction loss and knowledge-based loss. For the training of local models, the initial learning rate is 0.001, the maximum iteration number is 2000, the batch size is 1, and the dropout rate is 0.3.

E. Architecture of Local Models

To simplify the training of the proposed distributed framework, all the local models are with the same architecture in this

study. To be specific, each local model has two convolutional blocks (activation function is ReLU, batch normalization and dropout are employed in each convolutional block), two fully connected blocks (ReLU is selected as the activation function, batch normalization and dropout are also considered in each fully connected block), and one regression block. The final predictions of each local model could be obtained in the regression block.

F. Loss Function of Local Models

As shown in the right part of Fig. 3, the loss function of the local model has two components, i.e., the data prediction loss and knowledge-based loss. Different from the conventional data prediction loss, the novel loss function is to simultaneously minimize the mean square error (MSE) between the ground truths and predicted values, and the embedded prior physics knowledge, i.e., the equation of motion. In this manner, the novel loss function of the local model not only considers to minimize the data prediction error, but also brings the prior physics knowledge to make the created feature mapping satisfy the physical constraints of human movements. The knowledge-based loss component plays a regularization role, which could help strengthen the robustness of the created model due to the constraint of the prior physics knowledge. Moreover, the generalization of the local model also could be enhanced through encoding such physical information into the local model. The total loss function of the qth local model is

$$\begin{aligned} \mathcal{L}_{total} = \ & \mathcal{L}_{data} + \mathcal{L}_{know} \\ = \ & MSE(F,\theta) + \Gamma(F,\theta) \end{aligned} \tag{1}$$

where $\mathcal{L}_{data}$ is the MSE loss component, and $\mathcal{L}_{know}$ denotes the knowledge-based loss component, respectively.

Specifically, the MSE loss component is

$$MSE(F) = \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}(F_t^n - \hat{F}_t^n)^2 \tag{2}$$

$$MSE(\theta) = \frac{1}{T}\sum_{t=1}^{T}(\theta_t - \hat{\theta}_t)^2 \tag{3}$$

where F_t^n denotes the force of muscle n at time step t, θ_t represents the joint angle at time step t, while $\hat{F}_t^n$ and $\hat{\theta}_t$ are the corresponding predicted values. In addition, T denotes the number of samples, and N is the total muscle number, respectively.

Considering the equation of motion enables to reflect underlying relationships among the muscle forces and joint angle, we embed it into the knowledge-based loss component to strengthen the local model training:

$$\Gamma(F,\theta) = \frac{1}{T}\sum_{t=1}^{T}(M(\theta_t)\ddot{\theta}_t + C(\theta_t,\dot{\theta}_t) + G(\theta_t) - \tau_t)^2 \tag{4}$$

where $M(\theta_t)$ denotes the mass of the hand, which could be estimated based on the subject's body mass and height [8], $C(\theta_t,\dot{\theta}_t)$ is the Centrifugal and Coriolis force, $G(\theta_t)$ is the gravity, $\dot{\theta}_t$ and $\ddot{\theta}_t$ are the angular velocity and angular acceleration, and τ_t is the joint torque.

τ_t could be calculated by

$$\tau_t = \sum_{n=1}^{N} r_n F_t^n \tag{5}$$

where r_n denotes the moment arm of muscle n.

G. Aggregation of Local Models for Global Modelling

The proposed distributed framework establishes a number of local models, in which different types of neural networks could be freely selected for each of sub-domains. Such distributed structure enables easier parallelization of neural networks, which is quite essential in terms of achieving computational efficiency for the time-consuming deep learning training. After the local modelling for the separate representation of all the sub-domains, these local models should be integrated into a global model to obtain the final predictions.

Let $\mathbf{x}$, $\mathcal{N}^q(\mathbf{x})$ and Ω_q denote the input vector, output vector and parameter set of the qth local model, thus the outputs of the qth local model could be represented as

$$\mathbf{u}_{\Omega_q}(\mathbf{x}) = \mathcal{N}^q(\mathbf{x},\Omega_q) \tag{6}$$

Therefore, the integrated output of the global model is

$$\mathbf{u}_{\Omega}(\mathbf{x}) = \sum_{q=1}^{Q} w_q \mathbf{u}_{\Omega_q}(\mathbf{x}) \tag{7}$$

where w_q is the weight added to the qth local model:

$$w_q \propto \xi^q \tag{8}$$

where ξ^q is the total training error of the qth local model.

Accordingly, the weight of the qth local model could be calculated by

$$w_q = \frac{1}{\xi^q} \Big/ \sum_{q=1}^{Q}\frac{1}{\xi^q} \tag{9}$$

In this manner, all the local models are assigned weights based on training errors. Local models with small training errors could dominate the predictions of the global model, while the local models with larger training errors will impose less negative effects on the global performance. Therefore, the global model should be more robust and could better generalize to the unseen data.

H. Error Analysis

Total errors associated with local models could be summarized as the local optimization error ξ_{lo}^q, local approximation error ξ_{la}^q, and local generalization error ξ_{lg}^q, q = 1, …, Q. Specifically, ξ_{lo}^q is highly depending on the architecture and hyperparameters of the qth local model, such as depth, width, maximum iteration number, batch size and learning rate, etc. To reduce ξ_{la}^q, we should enhance the representation capability of the qth local model. The proposed physics-informed deep neural network should have better representation capability due to the embedded prior physics knowledge, thus its approximation error is smaller than state-of-the-art methods. However,

it may cause a relatively larger ξ_{lg}^{q}, which is known as the bias-variance trade-off in the machine/deep learning field [37], [38].

According to [39]–[41], all the three errors in each local model are inter-connected, it means that the total error of the proposed distributed framework should be smaller than the summation of the three errors of all the local models and errors of conventional machine/deep learning methods. Therefore, we have the following expression:

$$\xi_{total} \leqslant \sum_{q=1}^{Q} \xi_{lo}^{q} + \xi_{la}^{q} + \xi_{lg}^{q} \tag{10}$$

where ξ_{total} denotes the total error of the proposed distributed framework.

III. Performance Evaluation

In this section, the baseline methods and evaluation criteria are first detailed, and the effectiveness of the proposed distributed framework for musculoskeletal modelling is then evaluated, including comparisons with selected baseline methods, evaluation in intersession scenario, effects of the number of local models, effects of the number of training data, effects of the weighting strategy, and the time cost.

A. Baseline Methods and Evaluation Criteria

To comprehensively evaluate the feasibility of the proposed distributed framework, several methods are selected as baseline methods in the experiments.

1) CNN: CNN denotes the conventional CNN with the same network architecture with the local model of the proposed distributed framework. Similarly, during the training of CNN, its initial learning rate is 0.001, and the batch size is 1.

2) CNN-q: CNN-q is the qth local model utilized in the proposed distributed framework, its detailed parameter settings and training strategy have been given in Section II. CNN-q is trained using the data of the qth sub-domain.

3) Kn-CNN: Kn-CNN has the same architecture with CNN and CNN-q, its parameter settings and training strategy are same as CNN-q, and its loss function is with the knowledge-based loss component. Differently, it utilizes the whole data equally for the training purpose.

In the experiments, two criteria are considered to quantify the performance of all the methods, including the root mean square error (RMSE) and Pearson's correlation coefficient (CC). RMSE can be calculated by

$$\mathrm{RMSE} = \sqrt{\frac{1}{\tilde{T}} \sum_{\tilde{t}=1}^{\tilde{T}} \left(y_{\tilde{t}} - \hat{y}_{\tilde{t}}\right)^2} \tag{11}$$

where $y_{\tilde{t}}$ represents the ground truths (i.e., the true muscle forces and joint angles), $\hat{y}_{\tilde{t}}$ is the corresponding predicted values (i.e., the predicted muscle forces and joint angles), and $\tilde{T}$ and $\tilde{t}$ denote the number of data used in the experiments and index of data number, respectively.

CC is calculated by

$$CC = \frac{\sum_{\tilde{t}=1}^{\tilde{T}} \left(y_{\tilde{t}} - \overline{y_{\tilde{t}}}\right)\left(y_{\tilde{t}} - \overline{\hat{y}_{\tilde{t}}}\right)}{\sqrt{\sum_{\tilde{t}=1}^{\tilde{T}} \left(y_{\tilde{t}} - \overline{y_{\tilde{t}}}\right)^2} \sqrt{\sum_{\tilde{t}=1}^{\tilde{T}} \left(y_{\tilde{t}} - \overline{\hat{y}_{\tilde{t}}}\right)^2}} \tag{12}$$

where $\overline{y_{\tilde{t}}}$ and $\overline{\hat{y}_{\tilde{t}}}$ are the mean of the true muscle forces and joint angles, and the predicted muscle forces and joint angles.

B. Overall Comparisons with Baseline Methods

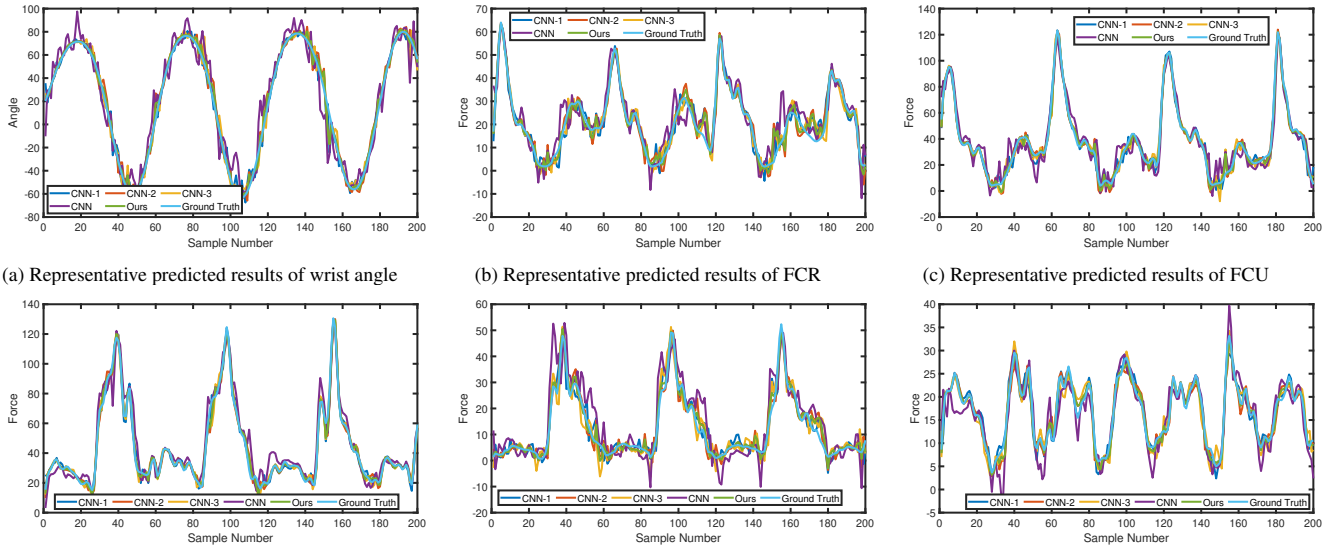
The overall comparisons among the proposed distributed framework and selected baseline methods are first demonstrated. It should be noted that we set the number of local models as three (i.e., $q = 1, 2, 3$) in this experiment, and effects of the number of local models on the performance is also investigated in the following experiment. The number of training data of the proposed distributed framework, CNN and Kn-CNN is 15000, and the number of training data of CNN-1, CNN-2 and CNN-3 is 5000. The number of testing data of all the methods is 2000.

Fig. 4 shows the representative predicted results of the wrist angle, muscle forces of ECRL, FCR, ECRB, FCU and ECU, and Table I lists the detailed quantitative comparison results across six subjects, including RMSE and CC. Observed from Fig. 4 and Table I, the proposed distributed framework achieves better performance than selected comparison methods, and its performance is more stable across six subjects. Because the proposed distributed framework employs a local-global distributed modelling mechanism, local models with smaller training errors dominate the global performance, which makes it more robust. In addition, the performance of all the three local models (i.e., CNN-1, CNN-2 and CNN-3) is better than that of CNN and Kn-CNN, because training deep learning models with smaller number (but sufficient) of data may significantly avoid the overfitting problem. In some situations, the performance of the proposed distributed framework is better than that of three local models, which demonstrates the feasibility and effectiveness of the local-global distributed modelling mechanism. Kn-CNN achieves better performance than that of CNN, indicating the effectiveness of the modified knowledge-based loss component in Kn-CNN. The performance of deep learning models with knowledge-based loss component is also enhanced through regularizing neural networks by the embedded prior physics knowledge.

Additionally, the pairwise analysis between the proposed distributed framework and each baseline method is carried out, where RMSE is selected as the response variable. A post-hoc analysis with Tukey's Honest Significant Difference test is employed. The significance level is $p < 0.05$. Fig. 5 depicts the average RMSEs and CCs across six subjects. According to Fig. 5, the performance of the proposed distributed framework is better.

C. Evaluation of Robustness and Generalization in Intersession Scenario

To verify the robustness and generalization of the proposed distributed framework in the intersession scenario, we select some of the collected data from four subjects as the training data, and the collected data from the remaining two subjects as the testing data. The number of training data and testing data of the proposed distributed framework, CNN, Kn-CNN, long short-term memory (LSTM), and CNN+LSTM is 15000 and



(a) Representative predicted results of wrist angle

(b) Representative predicted results of FCR

(c) Representative predicted results of FCU

(d) Representative predicted results of ECRL

(e) Representative predicted results of ECRB

(f) Representative predicted results of ECU

Fig. 4. Representative predicted results of the proposed distributed framework and selected baseline methods, including the predicted wrist angle, muscle forces of FCR, FCU, ECRL, ECRB, and ECU.

TABLE I
RMSEs AND CCs OF THE PROPOSED DISTRIBUTED FRAMEWORK AND BASELINE METHODS ACROSS SIX SUBJECTS

Variables	Methods	S1	S2	S3	S4	S5	S6	Variables	Methods	S1	S2	S3	S4	S5	S6
Angle	Ours	6.67/0.98	6.83/0.98	6.53/0.98	6.93/0.98	6.77/0.98	7.99/0.98	FCR	Ours	4.52/0.97	4.21/0.97	3.76/0.97	3.98/0.97	3.89/0.97	4.09/0.97
	CNN	13.03/0.91	14.75/0.91	11.97/0.93	13.69/0.91	12.95/0.92	13.08/0.90		CNN	7.15/0.92	8.20/0.89	9.13/0.90	8.82/0.91	8.55/0.92	9.06/0.91
	Kn-CNN	9.61/0.97	10.73/0.96	9.08/0.96	10.57/0.96	7.18/0.98	8.91/0.97		Kn-CNN	5.26/0.95	5.98/0.93	6.29/0.94	5.36/0.95	5.60/0.95	5.33/0.95
	CNN-1	7.95/0.98	8.55/0.97	6.79/0.98	9.28/0.96	7.59/0.98	8.99/0.97		CNN-1	4.62/0.96	4.69/0.97	3.98/0.97	5.25/0.95	4.21/0.96	4.95/0.96
	CNN-2	7.66/0.98	8.01/0.97	9.43/0.96	9.04/0.97	7.23/0.98	8.93/0.96		CNN-2	5.07/0.95	4.83/0.96	5.05/0.95	5.10/0.95	5.39/0.95	4.74/0.96
	CNN-3	7.52/0.98	8.12/0.98	9.67/0.96	9.03/0.97	7.06/0.98	9.27/0.96		CNN-3	4.75/0.96	4.72/0.96	5.23/0.95	4.62/0.96	4.75/0.96	5.21/0.96
FCU	Ours	4.91/0.98	4.62/0.98	4.56/0.98	4.54/0.98	6.11/0.97	4.09/0.98	ECRL	Ours	3.11/0.99	2.91/0.99	2.52/0.99	2.53/0.99	2.07/0.99	2.51/0.99
	CNN	8.11/0.95	8.67/0.94	9.52/0.90	10.73/0.91	11.33/0.91	8.65/0.94		CNN	6.89/0.96	8.17/0.95	6.59/0.97	7.55/0.95	7.99/0.95	7.69/0.95
	Kn-CNN	6.52/0.96	6.03/0.98	6.57/0.96	7.25/0.96	7.21/0.96	5.97/0.97		Kn-CNN	4.91/0.98	4.19/0.98	3.92/0.99	4.21/0.98	5.01/0.97	4.27/0.98
	CNN-1	5.36/0.97	5.59/0.97	4.90/0.98	5.43/0.97	5.10/0.98	5.71/0.98		CNN-1	3.46/0.99	3.56/0.99	3.13/0.99	3.63/0.99	3.15/0.99	3.46/0.99
	CNN-2	5.47/0.97	5.17/0.98	5.87/0.97	5.68/0.97	5.69/0.98	5.02/0.98		CNN-2	3.61/0.99	3.55/0.99	3.39/0.99	3.68/0.99	3.22/0.99	3.19/0.99
	CNN-3	5.29/0.98	5.22/0.98	5.96/0.97	5.99/0.97	7.07/0.97	4.23/0.98		CNN-3	3.76/0.99	3.38/0.99	3.58/0.99	3.32/0.99	3.69/0.99	3.98/0.99
ECRB	Ours	3.96/0.97	3.85/0.97	3.61/0.97	3.52/0.97	3.78/0.97	3.66/0.97	ECU	Ours	1.23/0.99	1.17/0.99	0.95/0.99	0.91/0.99	0.97/0.99	1.13/0.99
	CNN	6.83/0.90	7.83/0.87	6.96/0.88	8.02/0.90	8.21/0.89	7.09/0.90		CNN	2.17/0.95	2.85/0.94	3.02/0.92	3.57/0.92	3.25/0.93	3.77/0.91
	Kn-CNN	5.57/0.95	6.06/0.92	4.62/0.98	5.23/0.95	5.03/0.95	4.96/0.95		Kn-CNN	1.99/0.97	1.73/0.97	1.36/0.98	1.81/0.97	1.56/0.98	1.41/0.98
	CNN-1	4.32/0.96	4.64/0.95	4.09/0.97	4.50/0.95	4.23/0.96	4.40/0.96		CNN-1	1.57/0.98	1.49/0.98	1.44/0.98	1.42/0.98	1.40/0.98	1.37/0.98
	CNN-2	4.35/0.96	4.19/0.96	4.13/0.96	4.31/0.96	4.29/0.96	3.86/0.97		CNN-2	1.49/0.98	1.47/0.98	1.37/0.98	1.34/0.98	1.25/0.99	1.29/0.99
	CNN-3	4.40/0.96	4.42/0.96	4.56/0.95	4.37/0.96	4.09/0.97	4.11/0.96		CNN-3	1.55/0.98	1.39/0.99	1.34/0.98	1.22/0.99	1.38/0.98	1.46/0.98

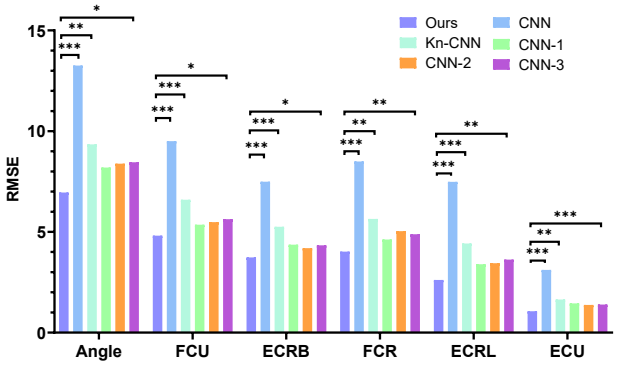
2000. Table II lists the comparison results, it could be found all the methods achieve worse performance than the results listed in Table I. Because the training data and testing data are from different subjects, their statistical characteristics may be different, degrading the predicted performance. Moreover, the proposed distributed framework could achieve the best performance among these methods in the intersession scenario, indicating its good robustness and generalization.

TABLE II
COMPARISON RESULTS AMONG THE PROPOSED DISTRIBUTED FRAMEWORK AND BASELINE METHODS IN INTERSESSION SCENARIO (RMSE)

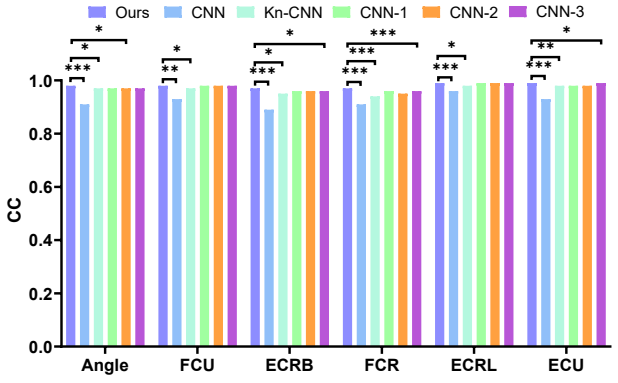
Methods	Angle	FCR	FCU	ECRL	ECRB	ECU
Ours ($q = 3$)	8.29	6.92	6.81	4.62	5.78	2.60
CNN	18.21	12.55	13.09	10.07	10.36	5.37
Kn-CNN	11.37	7.72	8.13	6.98	8.55	3.09
LSTM	16.39	10.38	15.32	8.39	9.92	5.79
CNN+LSTM	14.23	10.29	14.22	9.25	9.73	5.40

D. Effects of Number of Local Models

To further investigate the characteristics of the proposed distributed framework, we set different number of local models during the training phase and the maximum iteration is 2000. The total number of training data and testing data is 15000 and 2000, which are randomly selected from the collected data. For CNN, we use the whole data for the training purpose, i.e., the number of training data of CNN is 15000. For the



(a) Average RMSEs



(b) Average CCs

Fig. 5. Average RMSEs and CCs of the proposed distributed framework and baseline methods across six subjects. The proposed distributed framework achieves satisfactory performance. The significance level is set as 0.05 ($^{***}p < 0.001, ^{**}\ p < 0.01, and^{*} p < 0.05$).

proposed distributed framework, the whole data are divided into several sub-domains, the number of sub-domains equals to the number of local models. When the number of local models of the proposed distributed framework is 3, the number of training data of each local model is 5000. When the number of local models of the proposed distributed framework is 4, the number of training data is of CNN-1, CNN-2, CNN-3 and CNN-4 is 4000, 4000, 4000 and 3000. When the number of local models of the proposed distributed framework is 5, the number of training data is of CNN-1, CNN-2, CNN-3, CNN-4 and CNN-5 is 3000. Fig. 6 illustrates the changes of losses under different number of local models and CNN during the training processes. According to Fig. 6, separate losses of all the local models could converge quickly, while the losses of CNN are still divergent after 2000 iterations, indicating that the convergence speed of the proposed distributed framework is faster than that of CNN. Furthermore, we can find that losses of local models are more stable during the training process with the increase of the number of local models, because the number of training data in each sub-domain is smaller, the negative effects of unknown factors of specific sub-domains on the global performance are smaller, making the proposed distributed framework is less sensitive to unknown factors.

Table III lists the predicted results of the proposed distributed framework with various number of local models and CNN, it can be found that the proposed distributed framework

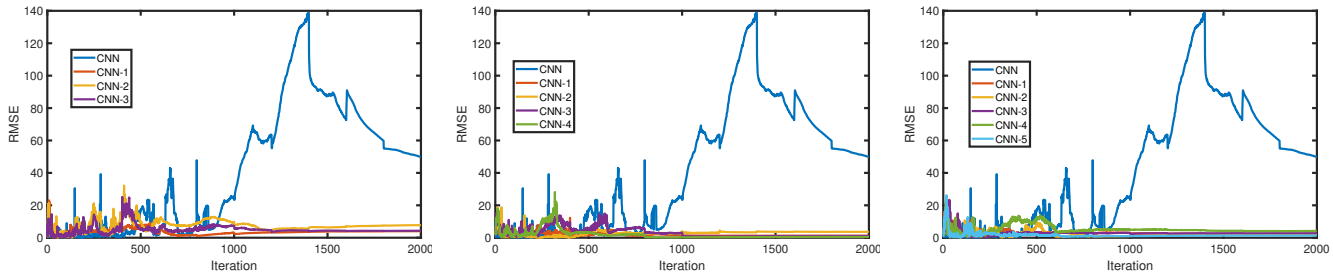
TABLE III
RMSE AND CC OF THE PROPOSED DISTRIBUTED FRAMEWORK WITH DIFFERENT NUMBER OF LOCAL MODELS AND CNN

RMSE						
Methods	**Angle**	**FCR**	**FCU**	**ECRL**	**ECRB**	**ECU**
Ours ($q = 3$)	6.83	4.21	4.63	2.91	3.85	1.16
Ours ($q = 4$)	6.52	3.76	4.55	2.52	3.61	0.96
Ours ($q = 5$)	6.93	3.98	4.51	2.53	3.52	0.91
CNN	14.74	8.20	8.66	8.17	7.83	2.82
CC						
Methods	**Angle**	**FCR**	**FCU**	**ECRL**	**ECRB**	**ECU**
Ours ($q = 3$)	0.98	0.97	0.98	0.99	0.97	0.99
Ours ($q = 4$)	0.98	0.98	0.98	0.99	0.97	0.99
Ours ($q = 5$)	0.98	0.97	0.98	0.99	0.98	0.99
CNN	0.92	0.89	0.94	0.96	0.87	0.95

achieves better performance than CNN in different scenarios. In addition, the predicted results of the proposed distributed framework with different number of local models are similar, indicating that its performance is not sensitive to the number of local models. Therefore, it could significantly reduce the time consumption of selecting appropriate number of local models when implementing the proposed distributed framework.

E. Effects of Number of Training Data

Aside from effects of the number of local models, effects of different training data sizes are also considered. We randomly select 10000 data pairs from the collected data for training and 2000 data pairs for testing, the number of local models in the proposed distributed framework is still set as 3. Table IV demonstrates the comparison results among the proposed distributed framework ($q = 3$), local models (i.e., CNN-1, CNN-2 and CNN-3) and CNN, in which the number of training data is from 1000 to 10000. According to Table IV, when the number of training data is small, the performance of the proposed distributed framework and CNN is comparable or the performance of the proposed distributed framework is even worse than CNN, and the performance of three local models is worse than CNN. Because the whole training data are divided into three sub-domains for training local models, and we cannot achieve good local models without enough training data, which also leads to the unsatisfactory performance of the proposed distributed framework. In addition, we can find that the performance of both the proposed distributed framework and local models becomes better than CNN with the increase of the number of training data. For example, when the number of training data is larger than 5000, the performance of both the proposed distributed framework and local models is much better than CNN. Furthermore, the experimental results also indicate that it is not very easy to train a good deep learning model when the number of training data is limited or large.



(a) Three local models (b) Four local models (c) Five local models

Fig. 6. Illustration of separate losses of the proposed distributed framework with different number of local models. The proposed distributed framework with various number of local models is with much faster convergence speed than CNN.

TABLE IV
COMPARISON RESULTS AMONG THE PROPOSED DISTRIBUTED FRAMEWORK, LOCAL MODELS AND CNN UNDER DIFFERENT NUMBER OF TRAINING DATA (RMSE)

Methods	No. Training Data	Angle	FCR	FCU	ECRL	ECRB	ECU
Ours ($q = 3$)	1000	19.85	8.88	10.52	8.80	7.37	**2.07**
	3000	17.72	8.07	8.15	12.31	6.69	3.18
	5000	**11.95**	6.24	**7.06**	**7.66**	**5.65**	2.82
	8000	12.01	**6.13**	7.36	8.15	6.25	2.93
	10,000	15.46	6.81	8.17	8.86	6.16	2.75
CNN-1	1000	24.26	10.78	12.30	9.99	8.57	**2.40**
	3000	20.67	8.89	9.02	13.20	7.29	3.41
	5000	**14.41**	**6.52**	8.12	8.39	6.46	3.03
	8000	15.59	7.01	**7.98**	**7.82**	**6.29**	2.44
	10,000	18.65	7.96	9.15	8.15	6.31	3.36
CNN-2	1000	27.81	10.09	14.99	11.93	8.81	3.47
	3000	15.34	8.07	8.73	10.52	6.90	**2.89**
	5000	15.02	7.82	8.56	10.15	7.26	3.83
	8000	**13.56**	**6.54**	**7.53**	**7.09**	**6.35**	2.96
	10,000	19.31	7.77	9.98	10.80	7.39	3.14
CNN-3	1000	24.53	9.78	12.88	11.97	8.89	2.62
	3000	15.69	7.72	8.81	11.14	6.99	3.58
	5000	14.49	7.91	8.23	7.39	6.57	2.64
	8000	16.63	8.02	10.26	11.24	7.83	3.62
	10,000	**14.46**	**7.10**	**8.12**	**6.75**	**6.42**	**2.56**
CNN	1000	17.92	9.13	**8.38**	**8.87**	6.33	**2.55**
	3000	**16.43**	**7.67**	8.79	10.29	**6.25**	2.98
	5000	30.17	12.67	19.60	21.99	12.75	5.79
	8000	33.34	14.77	18.97	20.89	13.45	4.79
	10,000	23.69	11.98	11.65	22.56	9.88	5.41

F. Effects of Weighting Strategy

Considering that some local models may be degraded by unknown factors during the training process, weighting strategy is employed in the proposed distributed framework to reduce and control negative effects of unknown factors on the global performance. The number of training data and testing data is 15000 and 2000, and Table V shows the comparison results. According to Table V, it could achieve better performance when weights are added to local models. Because local models with larger training errors are assigned smaller weights, which makes such local models impose smaller effects on the global performance, thus the proposed distributed framework is more robust especially in nonstationary scenarios.

G. Time Cost

The time costs among the proposed distributed framework and baseline methods are considered to verify the efficiency of the local-global distributed mechanism during the training phase. Table VI depicts the time costs of the proposed distributed framework with different number of local models and CNN when the number of training data is 8000, we can find that the time cost of the proposed distributed framework is much less than that of CNN. Moreover, the time cost is becoming less with the increase of the number of local models. Accordingly, we can find that with the increase of the number of training data, the proposed distributed framework should significantly outperform CNN in terms of time cost.

TABLE V
COMPARISON RESULTS OF THE PROPOSED DISTRIBUTED FRAMEWORK WITH AND WITHOUT WEIGHTING STRATEGY (RMSE)

Weights	Angle	FCR	FCU	ECRL	ECRB	ECU
With	6.61	4.06	4.55	2.73	3.89	1.08
Without	7.23	4.52	4.90	3.01	4.21	1.57

TABLE VI
COMPARISON RESULTS OF TIME COST OF THE PROPOSED DISTRIBUTED FRAMEWORK UNDER DIFFERENT NUMBER OF LOCAL MODELS AND CNN (MINUTE)

No. Training Data	Ours ($q = 3$)	Ours ($q = 4$)	Ours ($q = 5$)	CNN
8000	457	345	271	1388

IV. DISCUSSION

In this section, we discuss the potential advantages of the proposed distributed framework, such as its scalability and multimodal learning capability, limitations and future works.

A. Scalability in Training

In this study, we only use CNN as the baseline network of local models to demonstrate the feasibility and effectiveness of the proposed distributed framework, but it is actually scalable during the implementation. As illustrated in Fig. 7, various deep neural networks, such as CNN, recurrent neural network (RNN), long short-term memory (LSTM), stacked autoencoder

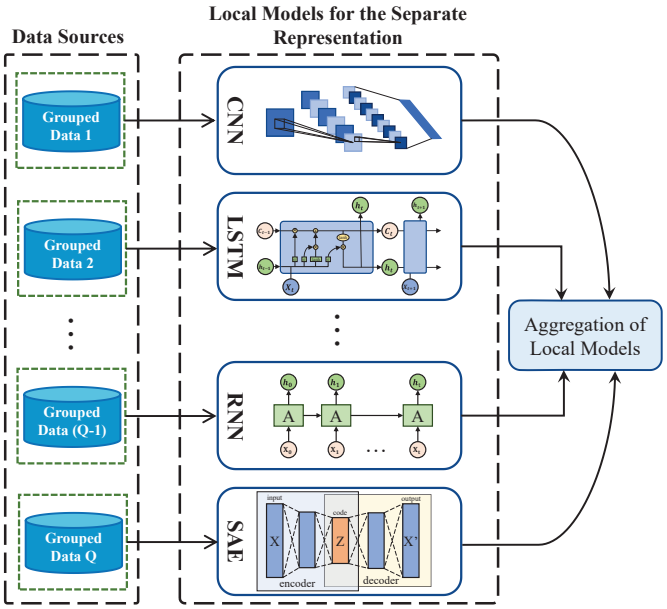


Fig. 7. The proposed distributed knowledge-driven deep learning framework is scalable. Local models could be flexibly designed with various depths, widths, activation functions, hyperparameters and types of neural networks depending on the statistical characteristics of the collected sEMG data and application requirements. The number of local models also could be set by the users or depending on the application requirements.

(SAE), and ResNet, etc., could be the candidates of local models. Additionally, the appropriate depth, width, activation function and hyperparameters also could be flexibly selected for the training of local models to achieve satisfactory performance for each of sub-domains. Such partial independence of local models training enables a set of parallel computations, which could significantly reduce the computational cost and memory requirements. Furthermore, it also makes the global performance of the proposed distributed framework not be seriously affected by specific local model(s). We can assign smaller weights to the local models with bad performance or directly remove them for global performance enhancement, making the proposed distributed framework more robust. Moreover, the training of local models also could be designed individually. Different local models could set various initial learning rates, batch sizes and stop criteria depending on the architecture of neural networks and statistical characteristics of the available training data in each of sub-domains.

B. Multimodal Learning Capability

State-of-the-art methods for musculoskeletal modelling are unimodal, which are developed only considering the single data source or modality for the defined tasks. However, doctors usually handle clinical data from multiple sources or modalities in real-world application scenarios, and thus the performance of unimodal musculoskeletal models fed with a single type of data is usually limited. In addition, Hendricks et al. [42] has demonstrated that data from multiple sources or modalities may be more useful than large number of data from single source or modality.

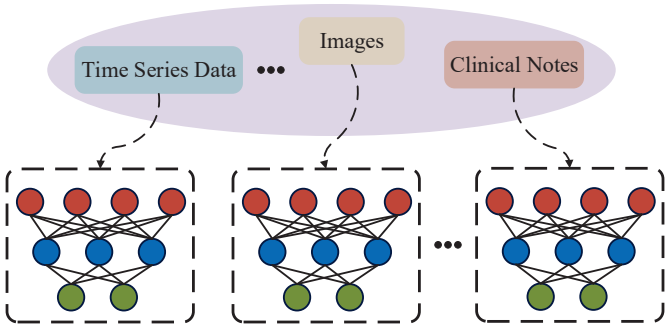


Fig. 8. The proposed distributed framework has the potential capability to learn from different sources or modalities in a unified framework, such as time series data, images and clinical notes, etc.

As shown in Fig. 8, the proposed distributed framework has the potential capability to learn from different sources or modalities in a unified framework, such as sEMG text data, images and data from various sensors. The proposed distributed framework is not with modality-specific biases (for example, RNN and LSTM are mainly for time series data, while CNN is with the great performance in image-related tasks), it thus may have satisfactory accuracy, robustness and generalization across a range of modalities. Boosting the multimodal learning capability of the proposed distributed framework also could enhance the personalized musculoskeletal modelling to obtain a wider and deeper understanding of musculoskeletal conditions of subjects [43], [44]. In the future, we will consider to design modified local model aggregation mechanism to make the proposed distributed framework be with excellent multimodal learning capability to assist the diagnostic, personalized treatment and clinical decision-making.

C. Limitations and Future Works

Although the proposed distributed framework could achieve satisfactory performance in several scenarios, there are still some limitations need to be addressed and proprieties need to be investigated in the future works. For example, how to choose appropriate data partition method for different tasks? We randomly select 6000 samples, and the number of local models is set as 3. Table VII details the comparison results between using k-means clustering method to partition the whole data (denoted by clustering) and the partition method mentioned in Section II-B (denoted by ours), we can find that the performance of the clustering-based distributed framework is worse. Fig. 9 visualizes the clustering results of the whole sEMG data, the number of data of one of the clusters is more than 5000 (denoted by yellow), while the number of data of the other two clusters is little, which leads to the corresponding two local models cannot be trained well, seriously degrading the global performance. Moreover, we only utilize the equation of motion in local models, more prior physics knowledge should be considered to further enhance the performance, such as the Hill muscle model [45]. In this study, all the local models are first trained locally, and then integrated into the global model, which actually makes the local-global modelling process relatively separate. In the future, we will consider to

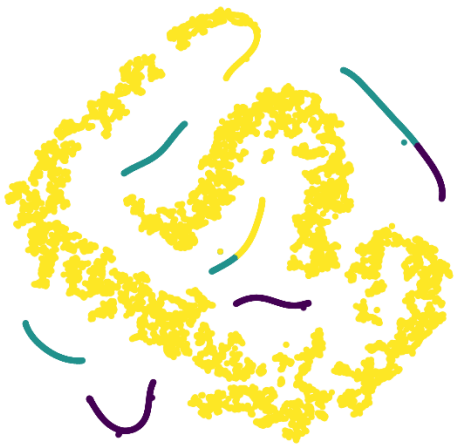


Fig. 9. Visualization of clustering results of sEMG data. The number of grouped data in the yellow cluster is much larger than the remaining two clusters, making the local models trained with data from these two clusters be with poor performance.

directly transfer the local model information to update the global model, such as the gradient information of all the local models [46].

TABLE VII
COMPARISON RESULTS OF THE PROPOSED DISTRIBUTED FRAMEWORK WITH DIFFERENT DATA PARTITION METHODS (RMSE)

Methods	Angle	FCR	FCU	ECRL	ECRB	ECU
Clustering	20.90	9.96	10.85	10.5367	7.93	3.07
Ours	7.02	4.15	5.11	3.89	3.82	1.17

V. CONCLUSION

In this study, a modified distributed framework based on physics-informed deep learning is designed to enhance the performance of the musculoskeletal modelling in nonstationary scenarios. The proposed distributed framework employs a local-global distributed modelling mechanism, where local models could flexibly select appropriate network types, architectures and hyperparameters depending on statistical characteristics of the collected data and application requirements, enabling locally to achieve better representation while preserving the global performance. Weighting strategy enables local models with better performance to dominate the global performance, and the negative effects of unknown factors can be reduced and controlled. In addition, the time cost of model training is also significantly reduced. Different from existing distributed modelling methods, prior physics knowledge is embedded into local models to regularise the loss functions of deep neural networks. The embedded prior physics knowledge not only helps enhance the robustness and generalization of the proposed distributed framework, but also makes the conventional "black-box" data-driven musculoskeletal modelling reflect underlying physical mechanisms.

Author Contributions: Conceptualization, Z.Q.Z., and J. Z; Methodology, Z.Q.Z., and J.Z.; Validation, J.Z and Z.L.R.; Writing-original draft preparation, J. Z, Q.L., and Z.L.R.; Writing-review and editing, Z.Q.Z., J.Z., ; Supervision and Funding acquisition, Z.Q.Z.. All authors have read and agreed to the published version of the manuscript.

Data Access Statement: The data presented in this study are available from the corresponding author upon request.

REFERENCES

[1] Y. Zhuang, S. Yao, C. Ma, and R. Song, "Admittance control based on EMG-driven musculoskeletal model improves the human–robot synchronization," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1211–1218, 2019.

[2] H. Zhang, F. Mo, L. Wang, M. Behr, and P.-J. Arnoux, "A framework of a lower limb musculoskeletal model with implemented natural proprioceptive feedback and its progressive evaluation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 8, pp. 1866–1875, 2020.

[3] H. Xu and A. Xiong, "Advances and disturbances in sEMG-based intentions and movements recognition: A review," *IEEE Sensors J.*, vol. 21, no. 12, pp. 13 019–13 028, 2021.

[4] P. Kang, J. Li, S. Jiang, and P. B. Shull, "Reduce system redundancy and optimize sensor disposition for EMG-IMU multi-modal fusion human-machine interfaces with XAI," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–9, 2023.

[5] A. Vijayvargiya, P. Singh, R. Kumar, and N. Dey, "Hardware implementation for lower limb surface EMG measurement and analysis using explainable AI for activity recognition," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–9, 2022.

[6] G. Durandau, D. Farina, and M. Sartori, "Robust real-time musculoskeletal modeling driven by electromyograms," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 3, pp. 556–564, 2018.

[7] F. S. Botros, A. Phinyomark, and E. J. Scheme, "Electromyography-based gesture recognition: Is it time to change focus from the forearm to the wrist?" *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 174–184, 2022.

[8] Y. Zhao, J. Zhang, Z. Li, K. Qian, S. Q. Xie, Y. Lu, and Z.-Q. Zhang, "Computational efficient personalised EMG-driven musculoskeletal model of wrist joint," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

[9] W. S. Burton II, C. A. Myers, and P. J. Rullkoetter, "Machine learning for rapid estimation of lower extremity muscle and joint loading during activities of daily living," *J. Biomech.*, vol. 123, p. 110439, 2021.

[10] C. P. Cop, A. C. Schouten, B. Koopman, and M. Sartori, "Electromyography-driven model-based estimation of ankle torque and stiffness during dynamic joint rotations in perturbed and unperturbed conditions," *J. Biomech.*, vol. 145, p. 111383, 2022.

[11] J. Weng, E. Hashemi, and A. Arami, "Adaptive reference inverse optimal control for natural walking with musculoskeletal models," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1567–1575, 2022.

[12] H. Su, W. Qi, Z. Li, Z. Chen, G. Ferrigno, and E. De Momi, "Deep neural network approach in EMG-based force estimation for human–robot interaction," *IEEE Trans. Artif. Intell.*, vol. 2, no. 5, pp. 404–412, 2021.

[13] S. Jiang, P. Kang, X. Song, B. P. Lo, and P. B. Shull, "Emerging wearable interfaces and algorithms for hand gesture recognition: A survey," *IEEE Rev. Biomed. Eng.*, vol. 15, pp. 85–102, 2022.

[14] T. Bao, S. Q. Xie, P. Yang, P. Zhou, and Z.-Q. Zhang, "Toward robust, adaptiveand reliable upper-limb motion estimation using machine learning and deep learning–A survey in myoelectric control," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 8, pp. 3822–3835, 2022.

[15] L. Rane, Z. Ding, A. H. McGregor, and A. M. Bull, "Deep learning for musculoskeletal force prediction," *Annu. Rev. Biomed. Eng.*, vol. 47, pp. 778–789, 2019.

[16] D. Xiong, D. Zhang, X. Zhao, and Y. Zhao, "Deep learning for EMG-based human-machine interaction: A review," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 3, pp. 512–533, 2021.

[17] G. Hajian and E. Morin, "Deep multi-scale fusion of convolutional neural networks for EMG-based movement estimation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 486–495, 2022.

[18] T. T. Dao, "From deep learning to transfer learning for the prediction of skeletal muscle forces," *Med. Biol. Eng. Comput.*, vol. 57, no. 5, pp. 1049–1058, 2019.

[19] W. Yang, D. Yang, Y. Liu, and H. Liu, "Decoding simultaneous multi-DOF wrist movements from raw EMG signals using a convolutional neural network," *IEEE Trans. Hum. Mach. Syst.*, vol. 49, no. 5, pp. 411–420, 2019.

[20] K.-T. Kim, C. Guan, and S.-W. Lee, "A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 94–103, 2020.

[21] H. Zhang, Y. Guo, and D. Zanotto, "Accurate ambulatory gait analysis in walking and running using machine learning models," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 1, pp. 191–202, 2020.

[22] K. Dinashi, A. Ameri, M. A. Akhaee, K. Englehart, and E. Scheme, "Compression of EMG signals using deep convolutional autoencoders," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 7, pp. 2888–2897, 2022.

[23] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.

[24] P. A. Reinbold, L. M. Kageorge, M. F. Schatz, and R. O. Grigoriev, "Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression," *Nat. Commun.*, vol. 12, no. 1, pp. 1–8, 2021.

[25] M. Sartori, D. G. Llyod, and D. Farina, "Neural data-driven musculoskeletal modeling for personalized neurorehabilitation technologies," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 5, pp. 879–893, 2016.

[26] C. Meng, S. Seo, D. Cao, S. Griesemer, and Y. Liu, "When physics meets machine learning: A survey of physics-informed machine learning," *arXiv preprint arXiv:2203.16797*, 2022.

[27] T. Bao, S. A. R. Zaidi, S. Xie, P. Yang, and Z.-Q. Zhang, "Inter-subject domain adaptation for CNN-based wrist kinematics estimation using sEMG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 1068–1078, 2021.

[28] X.-Y. Zhang, C.-L. Liu, and C. Y. Suen, "Towards robust pattern recognition: A review," *Proc. IEEE*, vol. 108, no. 6, pp. 894–922, 2020.

[29] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in nonstationary environments: A survey," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 12–25, 2015.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[31] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2018.

[32] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," *ACM Comput. Surv.*, vol. 55, no. 12, pp. 1–37, 2023.

[33] Zellers, Jennifer A and Parker, Sheridan and Marmon, Adam and Silbernagel, Karin Grävare, "Muscle activation during maximum voluntary contraction and m-wave related in healthy but not in injured conditions: Implications when normalizing electromyography," *Clin. Biomech.*, vol. 69, pp. 104–108, 2019.

[34] D. C. McFarland, E. M. McCain, M. N. Poppo, and K. R. Saul, "Spatial dependency of glenohumeral joint stability during dynamic unimanual and bimanual pushing and pulling," *J. Biomech. Eng.*, vol. 141, no. 5, 2019.

[35] D.G. Thelen, F.C. Anderson and S.L Delp, "Generating dynamic simulations of movement using computed muscle control," *J. Biomech.*, vol. 36, no. 3, pp. 321–328, 2003.

[36] D.G. Thelen and S.L Delp, "Using computed muscle control to generate forward dynamic simulations of human walking from experimental data," *J. Biomech.*, vol. 39, no. 6, pp. 1107–1115, 2006.

[37] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[38] P. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 525–536, 1998.

[39] K. Shukla, A. D. Jagtap, and G. E. Karniadakis, "Parallel physics-informed neural networks via domain decomposition," *J. Comput. Phys.*, vol. 447, p. 110683, 2021.

[40] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems," *Comput. Methods Appl. Mech. Eng.*, vol. 365, p. 113028, 2020.

[41] S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," *J. Comput. Phys.*, vol. 449, pp. 1–28, 2022.

[42] L. A. Hendricks, J. Mellor, R. Schneider, J.-B. Alayrac, and A. Nematzadeh, "Decoupling the role of data, attention, and losses in multimodal transformers," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 570–585, 2021.

[43] J. N. Acosta, G. J. Falcone, P. Rajpurkar, and E. J. Topol, "Multimodal biomedical AI," *Nat. Med.*, vol. 28, no. 9, pp. 1773–1784, 2022.

[44] A. Kline, H. Wang, Y. Li, S. Dennis, M. Hutch, Z. Xu, F. Wang, F. Cheng, and Y. Luo, "Multimodal machine learning in precision health: A scoping review," *NPJ Digit. Med.*, vol. 5, no. 1, p. 171, 2022.

[45] Y. Tan, Z. Fu, L. Duan, R. Cui, M. Wu, J. Chen, Y. Guo, J. Li, X. Guo, and H. Sun, "Hill-based musculoskeletal model for a fracture reduction robot," *Int. J. Med. Robot. Comp.*, vol. 17, no. 3, pp. 1–14, 2021.

[46] L. Zou, Z. Huang, X. Yu, J. Zheng, A. Liu, and M. Lei, "Automatic detection of congestive heart failure based on multiscale residual UNet++: From centralized learning to federated learning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.