Title
A multi-resolution physics-informed recurrent neural network: formulation and application to musculoskeletal systems

Permalink
https://escholarship.org/uc/item/9z37194j

Authors
Taneja, Karan
He, Xiaolong
He, QiZhi
et al.

Publication Date
2023-11-09

Peer reviewed

A Multi-Resolution Physics-Informed Recurrent Neural Network: Formulation and Application to Musculoskeletal Systems

Karan Taneja[a], Xiaolong He[b], QiZhi He[c] and Jiun-Shyan Chen[a*]

a Department of Structural Engineering, University of California San Diego, La Jolla, CA, USA
b ANSYS Inc., Livermore, CA, USA
c Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, Minneapolis, MN, USA

Abstract

This work presents a multi-resolution physics-informed recurrent neural network (MR PI-RNN), for simultaneous prediction of musculoskeletal (MSK) motion and parameter identification of the MSK systems. The MSK application was selected as the model problem due to its challenging nature in mapping the high-frequency surface electromyography (sEMG) signals to the low-frequency body joint motion controlled by the MSK and muscle contraction dynamics. The proposed method utilizes the fast wavelet transform to decompose the mixed frequency input sEMG and output joint motion signals into nested multi-resolution signals. The prediction model is subsequently trained on coarser-scale input-output signals using a gated recurrent unit (GRU), and then the trained parameters are transferred to the next level of training with finer-scale signals. These training processes are repeated recursively under a transfer-learning fashion until the full-scale training (i.e., with unfiltered signals) is achieved, while satisfying the underlying dynamic equilibrium. Numerical examples on recorded subject data demonstrate the effectiveness of the proposed framework in generating a physics-informed forward-dynamics surrogate, which yields higher accuracy in motion predictions of elbow flexion-extension of an MSK system compared to the case with single-scale training. The framework is also capable of identifying muscle parameters that are physiologically consistent with the subject's kinematics data.

Keywords: multi-resolution recurrent neural network, physics-informed parameter identification, musculoskeletal system, gated recurrent unit, fast wavelet transform

---

[*] Corresponding author
E-mail address: js-chen@ucsd.edu (Jiun-Shyan Chen)

1. Introduction

The prediction of the evolution of state variables in dynamical systems has been a vital component to several scientific applications such as biology, geophysics, earthquake engineering, solid mechanics, robotics, computer vision [1–7] etc. Black-box techniques based on data-driven mapping and development of parameterized multi-physics models describing the progression of the data have been previously utilized for making predictions on the states. This task continues to be an active area of research due to challenges on many fronts, such as, the quality and scarcity of relevant physical data, the dynamics and complexity of the system, and the reliability and accuracy of the prediction model.

On the other hand, the characterization of parameters in the multi-physics models of these dynamical systems is also critical [8–14]. The task is challenging due in parts to potential noise pollution captured by sensors in the system's measured data, as well as the potential of the parameter space being high-dimensional, leading to ill-posed problems that pose difficulties in numerical solutions. Standard optimization techniques such as genetic algorithms [15,16], simulated annealing [17], and non-linear least squares [18,19] have been employed for parameter identification, but can be computationally expensive and may not converge for ill-posed, non-convex optimization problems that are encountered while solving inverse problems on MSK systems [15,20].

In recent years, machine learning (ML) or deep-learning-based approaches have gained significant popularity for solving forward and inverse problems, attributed to their capability in effectively extracting complex features and patterns from data [21]. This has been successfully demonstrated in numerous engineering applications such as reduced-order modeling [22–26], and materials modeling [27–29], among others. Data-driven computing techniques that enforce constraints of conservation laws in the learning algorithms of a material database, have been developed in the field of computational mechanics [29–37]. More recently, physics-informed neural networks (PINNs) have been developed [11,38,39] to approximate the solutions of given physical equations by using neural networks (NNs). By minimizing the residuals of the governing partial differential equations (PDEs) and the associated initial and boundary conditions, PINNs have been successfully applied to solve forward problems [11,40,41], and inverse problems [11,38,42–44],

where the unknown system characteristics are considered trainable parameters or functions [38,45]. For biomechanics and biomedical applications [1,46–50], this method has been applied extensively along with other ML techniques [51,52]. These attempt to bridge the gap between ML-based data-driven surrogate models and the satisfaction of physical laws.

In this study, we focus on the application to musculoskeletal systems, aiming at utilizing non-invasive muscle activity measurements such as surface electromyography (sEMG) signals to predict joint kinetics or kinematics [1,18,19], which is of great significance to health assessment and rehabilitation purposes [15,16]. These sEMG signals can be used as control inputs to drive the physiological subsystems that are governed by parameterized non-linear differential equations, and thus form the forward dynamics problem. Given information on muscle activations, the joint motion of a subject-specific MSK system can be obtained by solving a forward dynamics problem. Data-driven approaches for motion prediction have also been introduced to directly map the input sEMG signal to joint kinetics/kinematics, bypassing the forward dynamics equations and the need for parameter estimation [26–30]. However, the resulting ML-based surrogate models lack interpretability and may not satisfy the underlying physics. Another challenge is that the sEMG signal usually exhibits a wide range of frequencies that are non-trivial for ML models [1] to map to the joint motion.

In our previous work [1], a physics-informed parameter identification neural network (PI-PINN) was proposed for the simultaneous prediction of motion and parameter identification with application to MSK systems. Using the raw transient sEMG signals obtained from the sensors and the corresponding joint motion data, the PI-PINN learned a forward model to predict the motion with identifying the parameters of the hill-type muscle models representing the contractile muscle-tendon complex. A feature-encoded approach was introduced to enhance the training of the PI-PINN, which yielded high motion prediction accuracy and identified system parameters within a physiological range, with only a limited number of training samples. However, this method relies on mapping in a feature domain constituted by Fourier and polynomial bases, which requires the input sEMG signal to span over the entire duration of the motion. Thus, it prevents *real-time* predictions as the signal is obtained from the sensor.

To enhance the predictive accuracy of the time-dependent signals, recurrent neural networks (RNNs) such as gated recurrent units (GRUs) [29,53] are utilized in this study to inform predictions

with the history information of the motion. To overcome the limitation of the size of the data and provide more information from the composite frequency bands in the signals, a multi-resolution based (MR) approach is proposed. In this approach, wavelets are used to decompose both the raw sEMG and joint motion signals into coarse-scale components at various frequency scales and the remaining fine-scale details. Using principles of the multi-resolution theory and transfer learning, multi-resolution training processes are repeated recursively from the coarse-scale to the full-scale in order to map the sEMG signal to the joint motion. Furthermore, gaussian noise is introduced to the recorded motion data used for training to enhance the robustness and generalizability of the model [29]. The trained model can be applied for real-time motion predictions given the raw sEMG signal obtained from the sensor.

This manuscript is organized as follows. Section 2 introduces the subsystems and mathematical formulations of MSK forward dynamics, followed by an introduction of the proposed multi-resolution PI-RNN framework for simultaneous motion prediction and system parameter identification in Section 3. The following sections verify the proposed framework using synthetic data and validate it by modeling the elbow flexion-extension movement using subject-specific sEMG signals and recorded motion data in Section 4 and 5, respectively. Concluding remarks and future work are summarized in Section 6.

2. Formulations for Muscle Mechanics and Musculoskeletal Forward Dynamics

This section provides a brief overview of muscle mechanics and forward dynamics of the human MSK system, with details in Appendix A and B. As depicted in Fig. 1, multiple subsystems within the MSK forward dynamics interact hierarchically: 1) the neural excitation $u(t)$ transforms into muscle activation $a(t)$ (activation dynamics); 2) Muscle activation drives muscle fibers to produce force F^{MT} (muscle-tendon (MT) contraction dynamics); 3) the resultant forces produce joint motion q (translation and rotation) of MSK systems, called the MSK forward dynamics [11,12,37].
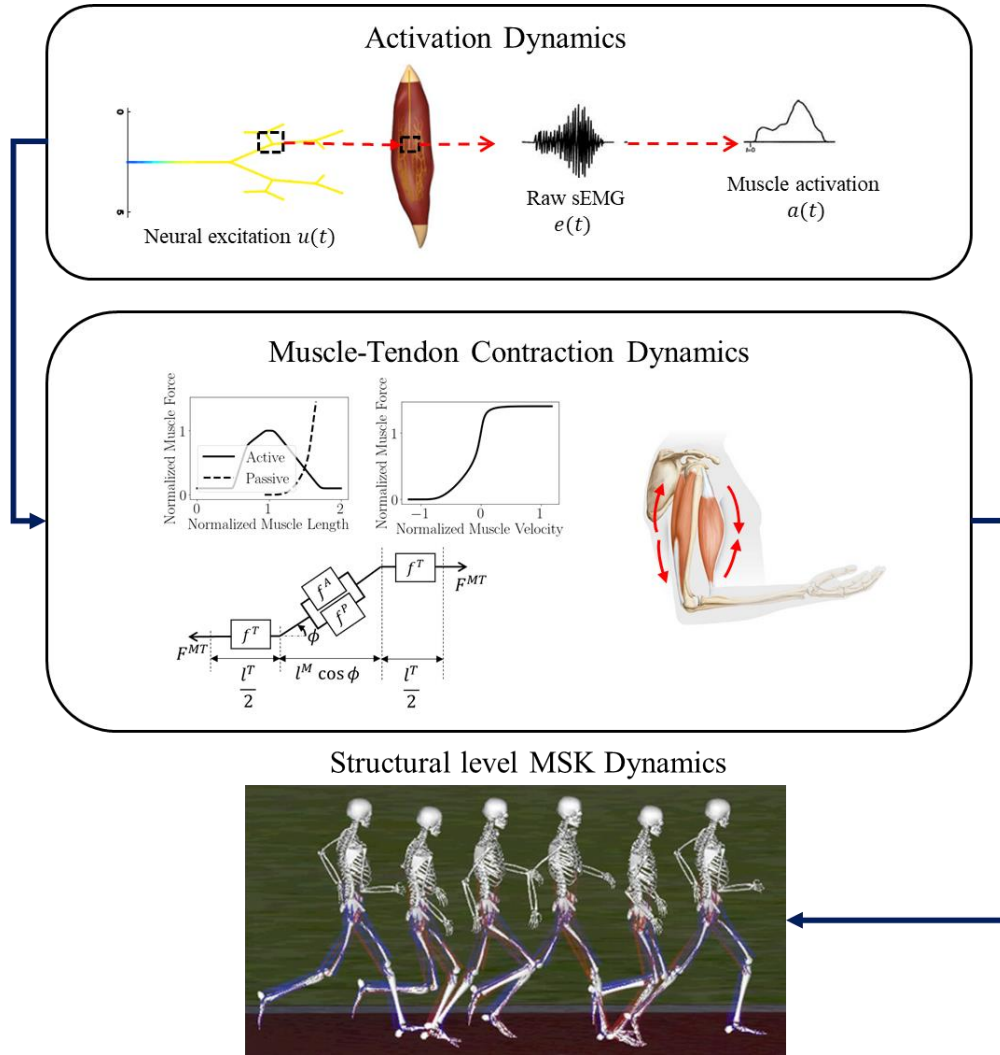
Fig. 1 The subsystems involved in the forward dynamics of an MSK system are depicted in this flowchart. Neural excitations are transmitted to muscle fibers (activation dynamics) that contract to produce force (muscle-tendon contraction dynamics). These forces generate torques at the joints (structural level MSK dynamics) leading to joint motion [1,54].

2.1 Neural Excitation-to-Activation Dynamics

While activations $a(t)$ in the muscle fibers can be obtained through a non-linear transformation on neural excitations $u(t)$, they are difficult to measure *in-vivo*. Therefore, the excitations are estimated from [15,16] the raw sEMG signals $e(t)$ considering an electro-mechanical delay:

$$u(t) = e(t - d). \tag{1}$$

where d measures the delay between the neural excitation originating and reaching the muscle group. The muscle activation signal $a(t)$ is then expressed as,

$$a(t) = \frac{\exp\big(Au(t)\big) - 1}{\exp(A) - 1} \tag{2}$$

where A is a shape factor. These activations initiate muscle fiber contraction leading to force production from the muscle group.

2.2 Muscle-Tendon Force Generation through Contraction Dynamics

(a)   (b)
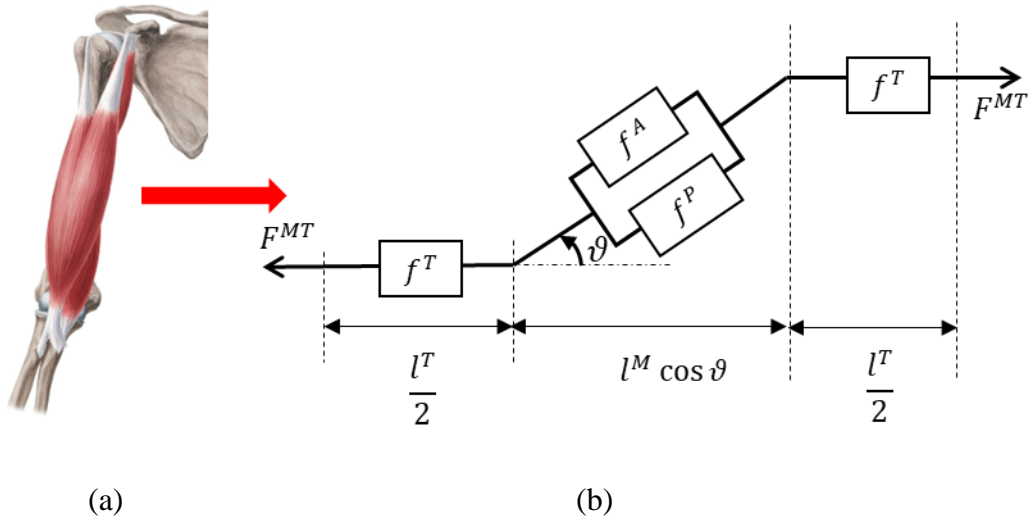
Fig. 2 A muscle-tendon complex in the arm modelled by a homogenized hill-type model where muscle group's in (a) are a homogenized muscle-tendon (MT) complex described by the model shown in (b).

Forces in the muscle-tendon (MT) complex are generated by the dynamics of MT contractions, where for structural length scale behaviour of the MT complex, homogenized hill-type muscle models are utilized (described in Appendix B) . Each muscle group can be characterized by a parameter vector,

$$\boldsymbol{\kappa} = [l_0^M, v_{max}^M, f_0^M, l_s^T, \vartheta_0], \tag{3}$$

containing constants such as the maximum isometric force in the muscle (f_0^M), the optimal muscle length (l_0^M) corresponding to the maximum isometric force, the maximum contraction velocity

(v_{max}^M), the slack length of the tendon (l_s^T), and the initial pennation angle (ϑ_0) [55,56]. The total force produced by the MT complex, F^{MT}, can be expressed as:

$$F^{MT}(a, \tilde{l}^M, \tilde{v}^M, \vartheta; \kappa) = F^M(a, \tilde{l}^M, \tilde{v}^M; \kappa) \cos \vartheta. \tag{4}$$

where a is the activation function in Eq. (2), $\tilde{l}^M$ is the normalized muscle length, $\tilde{v}^M$ is the normalized velocity of the muscle and ϑ is the current pennation angle. In this study, the tendon is assumed to be rigid $(l^T = l_s^T)$ which simplifies the MT contraction dynamics [57,58] accounting for the interaction of the activation, force length, and force velocity properties of the MT complex. More details can be found in Appendices A and B.

2.3 MSK Forward Dynamics of Motion

Body movement is the result of the force produced by actuators (MT complexes), converted to torques at the joints of the body, leading to rotation and translation of joints, which are considered as the generalized degrees of freedom of an MSK system (q). The dynamic equilibrium can be expressed as

$$I(q)\,\ddot{q} - T^{MT}(a, q, \dot{q}; \kappa) - E(q) = 0, \tag{5}$$

where $q, \dot{q}, \ddot{q}$ are the vectors of generalized angular motions, angular velocities, and angular accelerations, respectively; $E(q)$ is the torque from the external forces acting on the MSK system, e.g., ground reactions, gravitational loads etc.; $I(q)$ is the inertial matrix; T^{MT} is the torque from all muscles in the model calculated by $T^{MT}(a, q, \dot{q}; \kappa) = R(q)F^{MT}(a, q, \dot{q}; \kappa)$, where $R(q)$ are the moment arm's and $F^{MT}(a, q, \dot{q}; \kappa)$ are the forces from the MT complex. Given the muscle activation signals a, initial conditions and parameters of involved muscle groups κ, the generalized angular motions q and angular velocities $\dot{q}$ of the joints can be obtained by solving Eq. (5). An example of these vectors is shown in Section 4 and Appendix D.

3. Multi-Resolution Recurrent Neural Networks for Physics-Informed Parameter Identification

This section describes the recurrent neural network algorithms, followed by the physics-informed parameter identification that enables the development of a forward dynamics surrogate and simultaneous parameter identification. The employment of multi-resolution analysis based on fast

wavelet transform [59,60] for training data augmentation is then defined. The computational framework for multi-resolution recurrent neural network for physics-informed parameter identification is also discussed.

3.1 Recurrent Neural Networks and Gated Recurrent Units
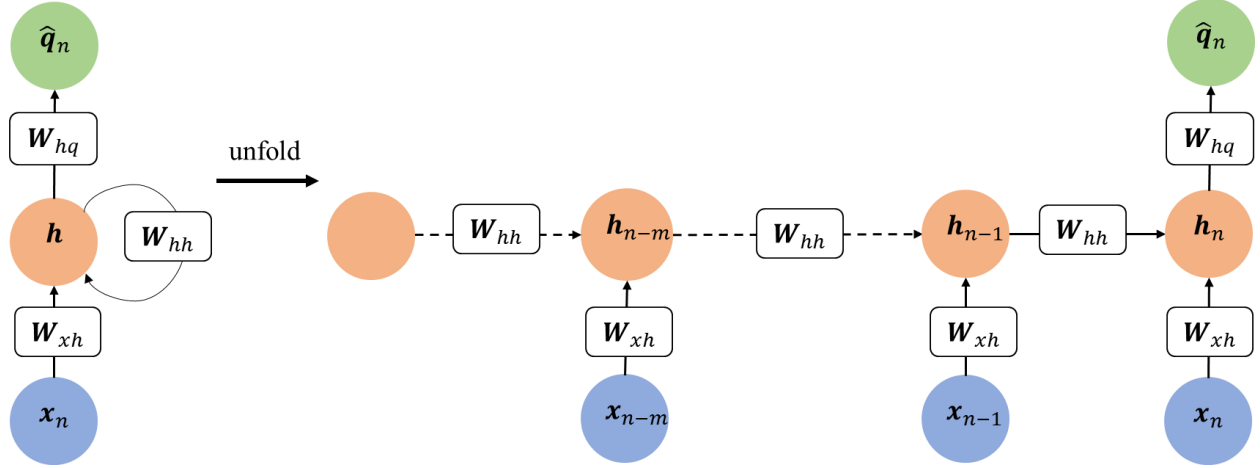
Fig. 3 Computational graph of a standard recurrent neural network using 'm' history steps for prediction.

The computational graph of a standard recurrent neural network (RNN) and its unfolded graph is shown in Fig. 3. The hidden state $\boldsymbol{h}$ allows for RNNs to learn important history-dependent features from the data in sequential time steps [29,53]. The unfolded graph shows the sharing of parameters across the architecture of the network, allowing for efficient training. The forward propagation of an RNN starts with an initial hidden state that embeds history-dependent features and propagates through all input steps. Considering an RNN with m history steps as shown in Fig. 3, the propagation of the hidden state can be expressed as follows [29].

$$\boldsymbol{h}_i = a_{tanh}(\boldsymbol{W}_{hh}\boldsymbol{h}_{i-1} + \boldsymbol{W}_{xh}\boldsymbol{x}_i + \boldsymbol{b}_h), \quad i = n - m, \dots, n \tag{6}$$

The hidden state at the final (current) step n is then used to inform the prediction.

$$\hat{\boldsymbol{q}}_n = \boldsymbol{W}_{hq}\boldsymbol{h}_n + \boldsymbol{b}_q \tag{7}$$

Here, a_{tanh} is the hyperbolic tangent function; $\boldsymbol{W}_{xh}, \boldsymbol{W}_{hh}$, and $\boldsymbol{W}_{hq}$ are the trainable weight coefficients; $\boldsymbol{b}_h$ and $\boldsymbol{b}_q$ are the trainable bias coefficients. The trainable parameters are shared

across all RNN steps. Let $\boldsymbol{x}_n = \left[t_n, e_n^1, \dots, e_n^{N_a}\right]$ be the current time and current sEMG data of the N_a muscle components and $\hat{\boldsymbol{q}}_n$ be the predicted joint motions at the current time t_n. Fig. 4(a) illustrates the computational graph of an RNN model trained to predict the motion at step n by using m history steps of $\boldsymbol{x}$ and $\boldsymbol{q}$ as well as the $\boldsymbol{x}$ at step n. The forward propagation is defined as

$$\boldsymbol{h}_i = a_{tanh}\left(\boldsymbol{W}_{hh}\boldsymbol{h}_{i-1} + \boldsymbol{W}_{xh}\boldsymbol{x}_i + \boldsymbol{W}_{qh}\boldsymbol{q}_i + \boldsymbol{b}_h\right), i = n - m, \dots, n - 1 \tag{8}$$

$$\boldsymbol{h}_n = a_{tanh}(\boldsymbol{W}_{hh}\boldsymbol{h}_{n-1} + \boldsymbol{W}_{xh}\boldsymbol{x}_n + \boldsymbol{b}_h), \tag{9}$$

$$\hat{\boldsymbol{q}}_n = \boldsymbol{W}_{h\hat{q}}\boldsymbol{h}_n + \boldsymbol{b}_q \tag{10}$$

with trainable parameters including the weight coefficients $\boldsymbol{W}_{hh}, \boldsymbol{W}_{xh}, \boldsymbol{W}_{qh}$ and $\boldsymbol{W}_{h\hat{q}}$ and bias coefficients $\boldsymbol{b}_h$ and $\boldsymbol{b}_q$. During training, the 'teacher-forcing' method is used where the measured motion data is given to the model in the history steps. In test mode, the model is fed back to the previous predictions as input to inform future predictions. The inputs received in this scenario could be quite different from those passed through in the training process, leading the network to make extrapolative predictions and therefore, accumulate errors which will pollute the predictions. To improve the testing performance and enhance model accuracy and robustness, a user-controlled amount of random Gaussian noise is added to the recorded motion data to introduce stochasticity so that the network can learn variable input conditions, resembling those in the test mode, see [29] for details.
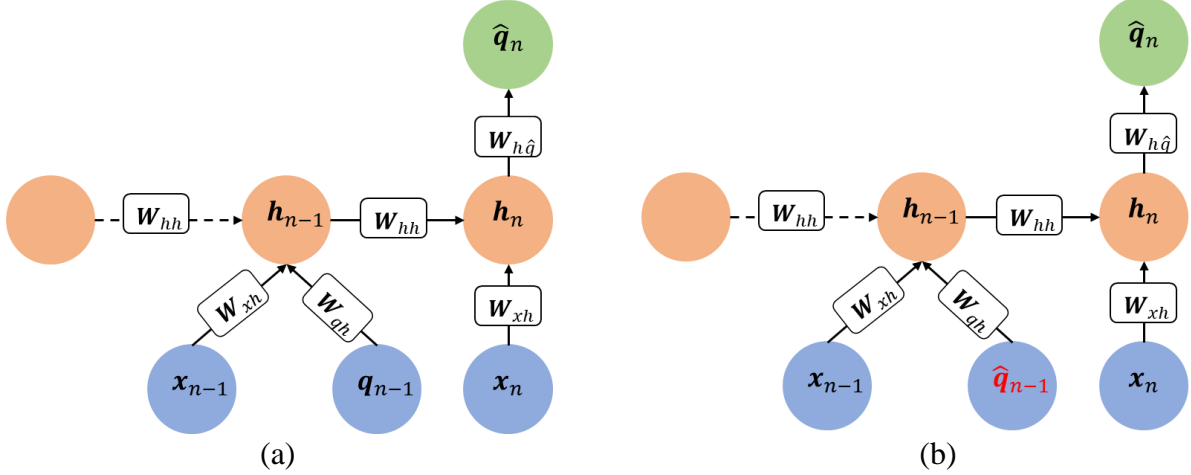
Fig. 4 An example computational graph of an RNN that uses one history step: (a) The train mode and (b) the test mode, where the motion predicted from the previous step is used as part of the input to predict motion at the current step.

Standard RNNs, however, have difficulties in learning long-term dependencies due to vanishing and exploding gradient issues arising from the recurrent connections. To mitigate these issues, gated recurrent units (GRUs) have been developed [48,50]. A standard GRU consists of a reset gate r_n, that removes irrelevant history information, an update gate u_n that controls the amount of history information that is passed to the next step, and a candidate hidden state $\widetilde{h}_n$ that is used to calculate the current hidden state h_n. Considering a GRU with m history steps, the forward propagation can be expressed as follows [29]:

$$
\begin{aligned}
\boldsymbol{r}_i &= a_\sigma\big(\boldsymbol{W}_{hr}\boldsymbol{h}_{i-1} + \boldsymbol{W}_{xr}\boldsymbol{x}_i + \boldsymbol{W}_{qr}\boldsymbol{q}_i + \boldsymbol{b}_r\big) \\
\boldsymbol{u}_i &= a_\sigma\big(\boldsymbol{W}_{hu}\boldsymbol{h}_{i-1} + \boldsymbol{W}_{xu}\boldsymbol{x}_i + \boldsymbol{W}_{qu}\boldsymbol{q}_i + \boldsymbol{b}_u\big) \\
\boldsymbol{z}_{(i,i-1)} &= \boldsymbol{r}_i \odot \boldsymbol{W}_{h\widetilde{h}}\boldsymbol{h}_{i-1} \\
\widetilde{\boldsymbol{h}}_i &= a_{tanh}\big(\boldsymbol{z}_{(i,i-1)} + \boldsymbol{W}_{x\widetilde{h}}\boldsymbol{x}_i + \boldsymbol{W}_{q\widetilde{h}}\boldsymbol{q}_i + \boldsymbol{b}_{\widetilde{h}}\big) \\
\boldsymbol{c}_{(i,i-1)} &= \boldsymbol{u}_i \odot \boldsymbol{h}_{i-1} \\
\widetilde{\boldsymbol{c}}_{(i,i)} &= \boldsymbol{u}_i \odot \widetilde{\boldsymbol{h}}_i \\
\boldsymbol{h}_i &= \boldsymbol{c}_{(i,i-1)} + \widetilde{\boldsymbol{h}}_i - \widetilde{\boldsymbol{c}}_{(i,i)} + \boldsymbol{b}_h
\end{aligned}
\tag{11}
$$

$$
\forall\, i = n - m, \dots, n - 1,
$$

$$r_n = a_\sigma(W_{hr}h_{n-1} + W_{xr}x_n + b_r)$$
$$u_n = a_\sigma(W_{hu}h_{n-1} + W_{xu}x_n + b_u)$$
$$z_{(n,n-1)} = r_n \odot W_{h\tilde{h}}h_{n-1} \tag{12}$$
$$\tilde{h}_n = a_{tanh}\big(z_{(n,n-1)} + W_{x\tilde{h}}x_n + b_{\tilde{h}}\big)$$
$$c_{(n,n-1)} = u_n \odot h_{n-1}$$
$$\tilde{c}_{(n,n)} = u_n \odot \tilde{h}_n$$
$$h_n = c_{(n,n-1)} + \tilde{h}_n - \tilde{c}_{(n,n)} + b_h$$

$$\hat{q}_n = W_{h\hat{q}}h_n + b_{\hat{q}} \tag{13}$$

where $\odot$ denotes the element-wise (Hadamard) product; $a_\sigma(\cdot)$ is the sigmoid activation function and $a_{tanh}(\cdot)$ is the hyperbolic tangent function; $W_{hr}, W_{xr}, W_{qr}, W_{hu}, W_{xu}, W_{qu}, W_{h\tilde{h}}, W_{x\tilde{h}}, W_{q\tilde{h}}$ and $W_{h\hat{q}}$ are the trainable weight coefficients; $b_r, b_u, b_{\tilde{h}}, b_h$ and $b_{\hat{q}}$ are the trainable bias coefficients. The current hidden state h_n is calculated by a linear interpolation between the previous hidden state h_{n-1} and the candidate hidden state $\tilde{h}_n$, based on the update gate u_n. The model is trained via the backpropagation through time algorithm applied to RNNs [21]. Training occurs by plugging in the measured motion data in history steps (shown in Fig. 5), known as the teacher forcing procedure [21]. For predictions, the prediction from the previous step is used to predict the current step. The addition of gaussian noise to measured data, as described before, is adopted in GRU models as well.
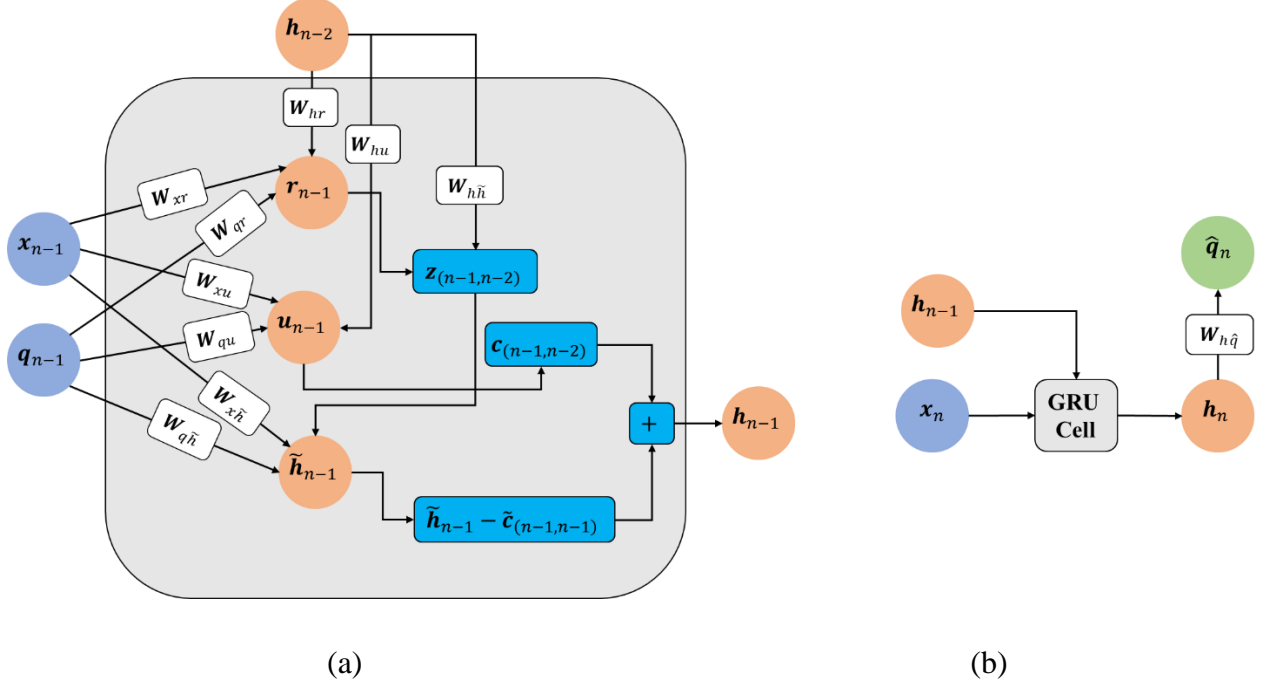
Fig. 5 An example computational graph of a GRU in *train* mode that uses one history step: (a) Starting with an initial or previously obtained hidden state (h_{n-2}), the main GRU cell takes the input x_{n-1} and motion q_{n-1} that are used to obtain the GRU hidden state h_{n-1} at step $n-1$ (Eq. (11)) and, (b) where the hidden state h_{n-1} is plugged back in to the GRU along with input x_n at step n to predict the motion $\hat{q}_n$ (Eq. (12)-(13)). The '+' cell produces an output (arrow pointing outwards) that is the summation of the inputs (arrows pointing into the cell).

3.2 Simultaneous Forward Dynamics Learning and Parameter Identification

With the governing equations for a general MSK forward dynamics (Section 2.1), the following parameterized ODE system is defined as

$$\mathcal{L}[q(t); \lambda] = s(t; \omega), \ \forall \ t \in \ (0, \text{T}], \ \mathcal{B}[q(0)] = g \ , \tag{14}$$

where the differential operator $\mathcal{L}[(\cdot); \lambda]$ is parameterized by a set of parameters λ. The right-hand side $s(t; \omega)$ is parameterised by ω. $\mathcal{B}[(\cdot)]$ is the operator for initial conditions, and g is the vector of prescribed initial conditions. To simplify notations, the ODE parameters are denoted by $\Gamma = \{\lambda, \omega\}$. The solution to the ODE system $q: [0, T] \to \mathbb{R}$ depends on the choice of parameters Γ.

Here, an RNN is used to relate data inputs containing discrete sEMG signals and discrete time from all the m previous history time-steps of a trial, $\cup_{i=n-m}^{n} x_i \in \mathbb{R}^{n_{in}}, m \in \mathbb{Z}^+$, to discrete joint motion data outputs at the current time-step, $q_n \in \mathbb{R}$, approximating the MSK forward dynamics. Let the training input at the ith history step be defined as $x_i = [t_i, e_i^1, \ldots, e_i^{N_a}]$, where t_i denotes the time at the ith time step, and $\{e_i^j\}_{j=1}^{N_a}$ denotes the sEMG signals of N_a muscle groups involved in the MSK joint motion at t_i. The motion at time step n, is then predicted using the training input from all the previous m steps using the RNN.

$$\hat{q}_n(\theta) = f_{RNN}(x_n, x_{n-1}, q_{n-1}, \ldots, x_{n-m}, q_{n-m}; \theta) \tag{15}$$

where f_{RNN} denotes RNN evaluations (depending on model chosen) discussed in Eq. (11)-(13). The optimal RNN parameters $\tilde{\theta}$ and the ODE parameters $\tilde{\Gamma}$ are obtained by minimizing the composite loss function J as follows,

$$\tilde{\theta}, \tilde{\Gamma} = \underset{\theta,\Gamma}{\operatorname{argmin}}(J) = \underset{\theta,\Gamma}{\operatorname{argmin}}(J_{data} + \beta J_{res}) \tag{16}$$

where β is the parameter to regularize the loss contribution from the ODE residual term in the loss function and can be estimated analytically [1]. The data loss is defined by,

$$J_{data} = \frac{1}{N_{data}} \sum_{\alpha=1}^{N_{data}} \|\hat{q}_\alpha(\theta) - q_\alpha\|_{L_2}^2 \tag{17}$$

where $\hat{q}_\alpha(\theta)$ is the predicted motion, and q_α is the recorded motion of MSK joints. In addition to training an MSK forward dynamics surrogate, the proposed framework aims to simultaneously identify important MSK parameters from the training data by minimizing residual of the governing equation of MSK system dynamics in Eq. (5).

$$J_{res} = \frac{1}{N_{data}} \sum_{\alpha=1}^{N_{data}} \|r(\hat{q}_\alpha(\theta); \Gamma)\|_{L_2}^2 \tag{18}$$

$$r(\hat{q}_\alpha(\theta); \Gamma) = \mathcal{L}[\hat{q}_\alpha(\theta); \lambda] - s(t_\alpha; \omega)$$

where $r(\hat{q}_\alpha(\theta); \Gamma)$ is the residual associated with Eq. (14) for the αth sample; $\Gamma = \{\lambda, \omega\}$ represents the ODE parameters relevant to the MSK system. The gradients of the network outputs with respect to the network parameters (θ), MSK parameters (Γ), and inputs are needed in the

loss function minimization in Eq. (16), which can be obtained efficiently by automatic differentiation [61]. The formulation in Eq. (15) is general such that more advanced RNN frameworks can be used such as the GRU described in Eq. (11)-(13).

3.3 Multi-Resolution Training with Transfer Learning

To improve the training efficiency of RNN for MSK applications with mixed-frequency sEMG input signals and low-frequency output joint motion, a multi-resolution decomposition of the training input-output data is introduced in Section 3.3.1, followed by the transfer learning based multi-resolution training protocols to be discussed in Section 3.3.2.

3.3.1 Wavelet based Multi-Resolution Analysis

Consider a sequence of nested subspaces $... \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(R)$ where $\cup_{j \in Z} V_j = L^2(R)$, and $\cap_{j \in Z} V_j = \emptyset$. Each subspace V_j of scale $[j]$ is spanned by a set of scaling functions $\phi_{j,k}(t)$, i.e.,

$$V_j = \left\{ \phi_{j,k}(t) \middle| \phi_{j,k}(t) = 2^{\frac{j}{2}} \phi(2^j t - k), k \in Z \right\}$$

Each subspace is related to the finer subspace through the law of dilation i.e., if $\phi(t) \in V_j$, then $\phi(2t) \in V_{j+1}, \forall j \in Z$. Translations of the scaling function span the same subspace, i.e., if $\phi(t) \in V_j$, then $\phi(t - k) \in V_j, \forall j, k \in Z$.

A mutually orthogonal complement of V_j in V_{j+1} is W_j, such that,

$$V_{j+1} = V_j \oplus W_j, \forall j \in Z \tag{19}$$

where $\oplus$ is a direct sum. This subspace W_j is spanned by a set of wavelet functions $\psi_{j,k}(t)$, i.e.,

$$W_j = \left\{ \psi_{j,k}(t) \middle| \psi_{j,k}(t) = 2^{\frac{j}{2}} \psi(2^j t - k), k \in Z \right\}$$

where $\psi(t)$ is the mother wavelet. It follows that,

$$\oplus_{j \in Z} W_j = L^2(R) \tag{20}$$

and therefore,

$$V_j = V_i \oplus \left(\oplus_{k=0}^{j-i-1} W_{i+k} \right), j > i. \tag{21}$$

The two-scale dilation and translation relations for the scaling functions can be written as

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} d_k \phi(2t - k). \tag{22}$$

Orthogonal wavelet functions can be obtained by imposing orthogonality conditions between scaling and wavelet functions in the frequency domain using Fourier transform,

$$\psi(t) = \sqrt{2} \sum_{k=-\infty}^{\infty} (-1)^{k-1} d_{-k-1} \phi(2t - k) \tag{23}$$

where d_k is the coefficient.

Orthogonal scaling functions can be constructed by choosing a candidate function $\phi^*(t)$ such that $\phi^*(t)$ have reasonable decay and a finite support. In addition, $\int \phi^*(t)dt \neq 0$. It should also satisfy the two-scale relation,

$$\phi^*(t) = \sum_k p_k \phi^*(2t - k), k \in \mathcal{Z}. \tag{24}$$

With these, an orthogonal scaling function $\phi(t)$ can be expressed in terms of $\phi^*(t)$ as

$$\phi(t) = \sum_{k=-\infty}^{\infty} a_k \phi^*(t - k). \tag{25}$$

It is then possible to define the scaling function at the coarse scale in terms of the scaling function at the fine scale and the wavelet functions at the coarser scale,

$$\phi(2t - l) = \sum_{k=-\infty}^{\infty} d_{l-2k} \phi(t - k) + \sum_{k=-\infty}^{\infty} h_{l-2k} \psi(t - k), l \in \mathcal{Z}. \tag{26}$$

Any function can be approximated at scale $[j]$ by using $\phi_{j,k}$ as a basis as well as using its coarse scale $[j - 1]$ representation and details at the coarse scale, i.e.,

$$P_j f = \sum_{k=-\infty}^{\infty} S_k^{[j]} \phi_{j,k} = P_{j-1}f + H_{j-1}f \qquad (27)$$

$$= \sum_{k=-\infty}^{\infty} S_k^{[j-1]} \phi_{j-1,k} + \sum_{k=-\infty}^{\infty} T_k^{[j-1]} \psi_{j-1,k}$$

where P_j and H_j are the operators projecting f onto the subspaces V_j and details of f at scale $[j]$ in the orthogonal subspace W_j, respectively. $S_k^{[j]}$ and $T_k^{[j]}$ are the corresponding basis coefficients at the coarse scale $[j]$. While the example shown here is for a one-dimensional case, this multi-resolution representation can be extended to multi-dimensions.

3.3.2 Multi-Resolution Data Representation and Training Protocols

In this approach, a given signal $f(t)$ is represented using the multi-resolution scaling functions and wavelets. A scale $[j]$ representation of signal $f(t)$ can be obtained from the scale $[r]$ $(j > r)$ representation with the addition of wavelet components (high frequency components) of the scales higher than $[r]$, using the discrete wavelet transform modified from Eq. (27),

$$P_j f(t) = P_r f(t) + \sum_{b=r}^{j-1} H_b f(t) = \sum_{k=-\infty}^{\infty} S_k^{[r]} \phi_{r,k}(t) + \sum_{b=r}^{j-1} \sum_{k=-\infty}^{\infty} T_k^{[b]} \psi_{b,k}(t) \qquad (28)$$

where P_r is the projection operator at scale $[r]$ and H_b are the wavelet projectors of the signal that are added from scale $[r]$ to scale $[j-1]$ to reconstruct the signal at scale $[j]$; $S_k^{[r]}$ and $T_k^{[b]}$ are the scaling and wavelet function's coefficients, obtained by the orthogonality condition as given in Section 3.3.1.

Using the Wavelet transform to represent a time series under multiple resolutions offers advantages for feature extraction from signals. Compared to the Fourier transform which offers only localization in the frequency domain, the Wavelet transform provides both frequency and time domain localization, making it more suitable for time history (or sequence) learning algorithms such as the standard RNN and its enhanced variant GRU. More specifically, one can enhance training efficiency by using a sequential training strategy for the time-history input (sEMG) and output (joint motion) data. Applying the Fast Wavelet Transform [57,58] to obtain the input and output data from low to high resolutions results in better generalization performance of the RNN

16

trained to map from sEMG signals to joint motion time history as described below. The second order Daubechies wavelets are used in this work.

Here we consider a general MSK system described in Section 2. The original unfiltered data is denoted as scale [0], which will be decomposed into a sequence of lower scales $[-j], j \in \mathbb{Z}^+$ for multi-resolution training.

Let $\boldsymbol{D}^{[0]}$ be the input training data at the full-scale $(j = 0)$ of the raw signals i.e.,

$$\boldsymbol{D}^{[0]} = \left[\boldsymbol{x}_1^{[0]}, \boldsymbol{x}_2^{[0]}, \dots, \boldsymbol{x}_{N_{data}}^{[0]} \right], \tag{29}$$

$$\boldsymbol{x}_i^{[0]} = \left[t_i, e_i^{1[0]}, \dots, e_i^{Na[0]} \right].$$

and the motion of joints of the MSK system at the ith time-step at the full-scale $(j = 0)$ is $\boldsymbol{q}_i^{[0]}$ such that the array of the unfiltered motion data for the duration of the motion is $\boldsymbol{q}^{[0]} = \left[\boldsymbol{q}_1^{[0]}, \boldsymbol{q}_2^{[0]}, \dots \boldsymbol{q}_{N_{data}}^{[0]} \right]$.

From MR theory, subtracting details from the fine scale representations at the full-scale of the signal, i.e., [0], results in a course scale representation of the signal at scale $[-k], k = 1, \dots j$. The projected training data at coarse scale [-j] is defined as

$$\boldsymbol{D}^{[-j]} = \left[\boldsymbol{x}_1^{[-j]}, \boldsymbol{x}_2^{[-j]}, \dots, \boldsymbol{x}_{N_{data}}^{[-j]} \right], \tag{30}$$

where N_{data} is the total number of data points and

$$\boldsymbol{x}_i^{[-j]} = \left[t_i, e_i^{1[-j]}, \dots, e_i^{Na[-j]} \right], i=1\dots, N_{data} \tag{31}$$

is the input data of scale [-j] at time step i. The motion of the MSK joints at the ith time-step at the scale $[-j]$ is $\boldsymbol{q}_i^{[-j]}$. The data sets for a representative muscle group 'MT', $e_i^{MT[-j]}$ and motion $\boldsymbol{q}_i^{[-j]}$, are obtained from the original raw data $e_i^{MT[0]}$ and $\boldsymbol{q}_i^{[0]}$ by wavelet projection using Eq. (27), that is,

$$e^{MT[-j]}(t) \equiv P_j e^{MT[0]}(t) = P_{j-1} e^{MT[0]}(t) + H_{j-1} e^{MT[0]}(t)$$

$$= e^{MT[0]}(t) - \sum_{b=0}^{j-1} H_b e^{MT[0]}(t) \tag{32}$$

$$q^{[-j]}(t) \equiv P_j q^{[0]}(t) = P_{j-1} q^{[0]}(t) + H_{j-1} q^{[0]}(t) = q^{[0]}(t) - \sum_{b=0}^{j-1} H_b q^{[0]}(t)$$

where P_j and H_j are the projection operators in multi-dimensions. Hence, datasets that contain lower resolution representations of the original signal at scales [0] can be expressed as:

$$D^{[-j]} \subset D^{[-j+1]} \subset \cdots D^{[-1]} \subset D^{[0]} \tag{33}$$

$$q^{[-j]} \subset q^{[-j+1]} \subset \cdots q^{[-1]} \subset q^{[0]}$$

where $q^{[-j]} = \left[q_1^{[-j]}, q_2^{[-j]}, \ldots, q_{N_{data}}^{[-j]} \right]$.

Instead of learning the signal mapping from input original raw sEMG data $D^{[0]}$ to motion data $q^{[0]}$, we initiate learning the mapping by starting from a *coarse* scale representation of the input-output data at scale $[-j]$ and map $D^{[-j]}$ to $q^{[-j]}$. For multi-resolution RNN, the initial learning starts from the coarsest scale $[-j]$ as follows:

$$h_i^{[j]} = a_{tanh} \left(W_{hh}^{[-j]} h_{i-1}^{[-j]} + W_{xh}^{[-j]} x_i^{[-j]} + W_{qh}^{[-j]} q_i^{[-j]} + b_h^{[-j]} \right), \tag{34}$$

$$\forall i = n - m, \ldots, n - 1$$

$$h_n^{[-j]} = a_{tanh} \left(W_{hh}^{[-j]} h_{n-1}^{[-j]} + W_{xh}^{[-j]} x_n^{[-j]} + b_h^{[-j]} \right), \tag{35}$$

$$\hat{q}_n^{[-j]} = W_{h\hat{q}}^{[-j]} h_{n-1}^{[-j]} + b_q^{[-j]}. \tag{36}$$

At the next finer scale $[-j+1]$, the weights at scale $[-j]$ (using an early stopping [62]) are used as the initial values for $W_{hh}^{[-j+1]}$, $W_{xh}^{[-j+1]}$, $W_{qh}^{[-j+1]}$, $W_{h\hat{q}}^{[-j+1]}$, $b_h^{[-j]}$, $b_q^{[-j]}$, similar to the concept of transfer learning[63].

Similarly, for multi-resolution GRU, the initial learning starts from the coarsest scale $[-j]$ as described in Appendix C. The same procedures to transfer the NN parameters in Eq.(34)-(36) are repeated with $[-j] \rightarrow [-j+1]$ until it reaches scale [0]. To enhance model accuracy and robustness, variations based on Gaussian noise are added to the motion data in each sequential step, as suggested by [29]. The sequential MR training process is described in Algorithm 1.

Algorithm 1: Sequential Multi-Resolution PI-RNN training process.

Step1: Initialize parameters.

$$\boldsymbol{\theta} = \boldsymbol{\theta}^0, \boldsymbol{\Gamma} = \boldsymbol{\Gamma}^0$$

Step 2: Sequential learning through parameter transfer from coarse-scale to fine-scale.

For $l = 0 \rightarrow j$

1. Train the RNN on the dataset $\boldsymbol{D}^{[-j+l]}$ by calculating the predictions as

$$\hat{\boldsymbol{q}}_n^{[-j+l]}(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = f_{RNN}\left(\boldsymbol{x}_n^{[-j+l]}, \boldsymbol{x}_{n-1}^{[-j+l]}, \boldsymbol{q}_{n-1}^{[-j+l]}, \dots, \boldsymbol{x}_{n-m}^{[-j+l]}, \boldsymbol{q}_{n-m}^{[-j+l]}; \boldsymbol{\theta}, \boldsymbol{\Gamma}\right)$$

$$m \rightarrow \# \text{ of history steps}$$

$$n \rightarrow \text{current time step}$$

$$\widetilde{\boldsymbol{\theta}}, \widetilde{\boldsymbol{\Gamma}} = \underset{\boldsymbol{\theta}, \boldsymbol{\Gamma}}{\operatorname{argmin}}(J) = \underset{\boldsymbol{\theta}, \boldsymbol{\Gamma}}{\operatorname{argmin}}\left(J_{data}(\boldsymbol{\theta}) + \beta J_{res}(\boldsymbol{\theta}, \boldsymbol{\Gamma})\right)$$

2. $\boldsymbol{\theta} = \widetilde{\boldsymbol{\theta}}, \boldsymbol{\Gamma} = \widetilde{\boldsymbol{\Gamma}}$

4. Verification Example

For verification of the proposed MR PI-RNN framework, an elbow flexion-extension model [1] and synthetic sEMG signals with gaussian noise and associated motion responses were considered. The flowchart of the proposed computational framework for simultaneous forward dynamics prediction and parameter identification of MSK parameters is shown in Fig. 6.

The model contained two rigid links corresponding to the upper arm and forearm with lengths l_{ua} and l_{fa}, respectively. They were connected at a hinge resembling the elbow joint "A", while the upper arm link was fixed at the top joint "B", and the biceps (*Bi*) and triceps (*Tri*) muscle-tendon complexes (modeled by Hill-type models with parameters $\boldsymbol{\kappa}_{Bi}$ and $\boldsymbol{\kappa}_{Tri}$) were represented by the lines connecting the links, as shown in Fig. 6. The degree of freedom of the model was the elbow flexion angle q. The mass in the forehand was assumed to be concentrated at the wrist location, hence, a mass m_{fa} was attached to one end of the forearm link with a moment arm l_{fa} from the elbow joint. Tendons were assumed as rigid [58] for ease of computation.
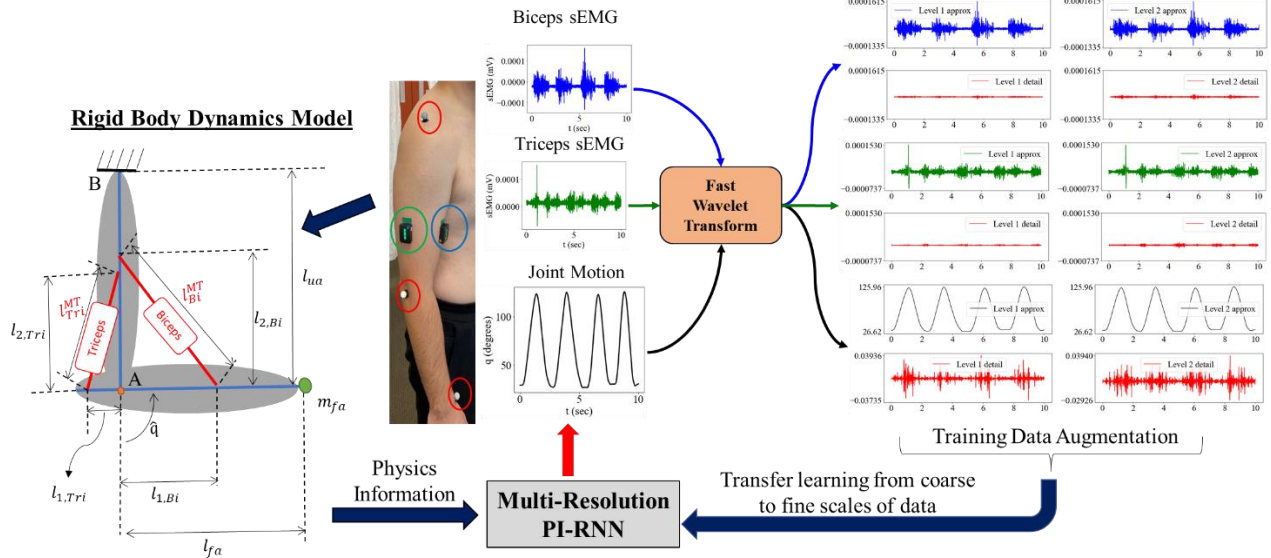
Fig. 6 An overview of the application of this framework to the recorded motion data. The location of motion capture markers is circled in red and the sEMG sensors on Biceps and Triceps muscle groups in blue and green, respectively. The simplified rigid body model was used in the forward dynamics equations within the framework with appropriately scaled anthropometric properties (for geometry) and physiological parameters (for muscle-tendon material models). The raw sEMG signals were mapped to the target angular motion of the elbow and used to simultaneously characterize the MSK system using the proposed Multi-Resolution PI-RNN framework.

The equation of motion for this rigid body system is given in Appendix D. Given the synthetic sEMG signals $(e^{Bi}(t), e^{Tri}(t))$, the initial conditions $q(0) = \frac{\pi}{6}$ radians and $\dot{q}(0) = 0$ radians/ sec and the parameters in Table 1, the motion of the elbow joint, q, can be obtained by solving the MSK forward dynamics problem using an explicit Runge-Kutta scheme, implemented in Python's SciPy library [64].

Table 1: Parameters involved in the forward dynamics setup of elbow flexion-extension motion.

Parameter	Type	Value	Parameter	Type	Value
$l_{0,Bi}^{M}$	Biceps Muscle Model	0.6 m	m_{fa}	Equation of motion	1.0 kg
$v_{max,Bi}^{M}$	Biceps Muscle Model	6 m/sec	l_{ua}	Geometric	1.0 m
$f_{0,Bi}^{M}$	Biceps Muscle Model	300 N	l_{fa}	Geometric	1.0 m
$l_{s,Bi}^{T}$	Biceps Muscle Model	0.55 m	$l_{1,Bi}$	Geometric	0.3 m
ϕ_{Bi}	Biceps Muscle Model	0.0 radians	$l_{2,Bi}$	Geometric	0.8 m
$l_{0,Tri}^{M}$	Triceps Muscle Model	0.4 m	$l_{1,Tri}$	Geometric	0.2 m
$v_{max,Tri}^{M}$	Triceps Muscle Model	4 m/sec	$l_{2,Tri}$	Geometric	0.7 m
$f_{0,Tri}^{M}$	Triceps Muscle Model	300 N	d	Activation Dynamics	0.08 sec
$l_{s,Tri}^{T}$	Triceps Muscle Model	0.33 m	A	Activation Dynamics	0.2
ϕ_{Tri}	Triceps Muscle Model	0.0 radians			
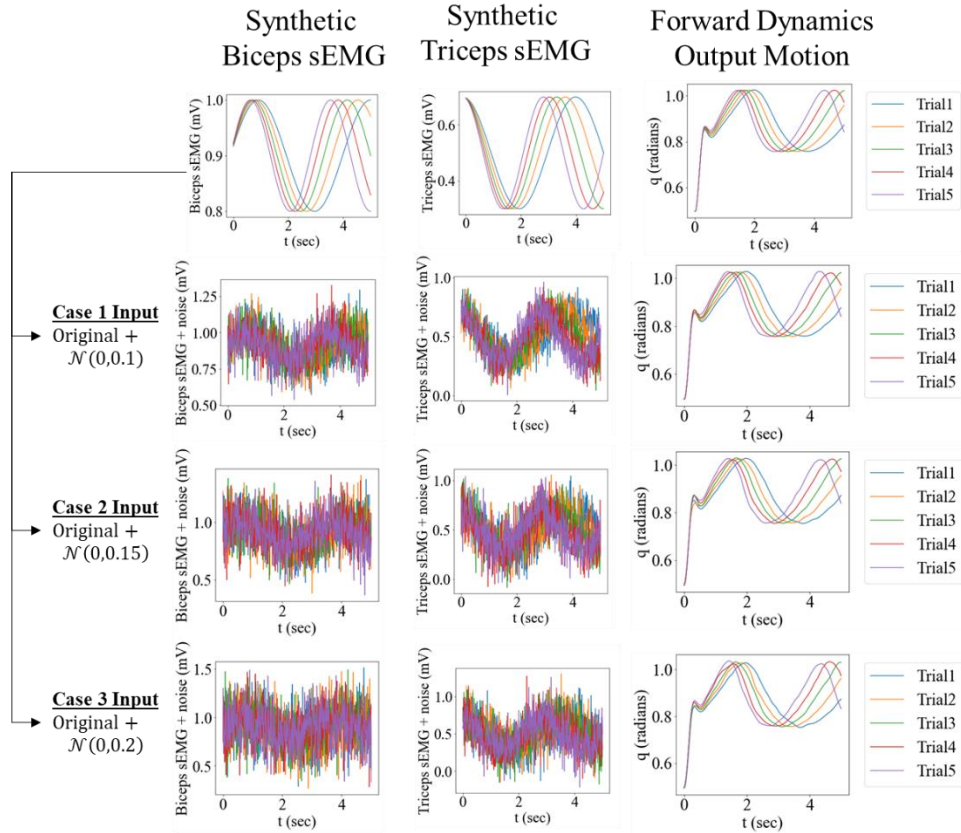
Fig. 7 The original 'noiseless' input-output data set with the synthetic biceps and triceps sEMG signals having variations in frequency for five trials are shown at the top. Increasing levels of noise are added to develop 3 cases of synthetic mixed frequency input sEMG, from which corresponding output motions are solved, using the forward dynamics equations. To verify the MR framework, these three cases with their respective mixed frequency input data are then mapped to their corresponding motion data.

To verify and check the robustness of the MR framework to different levels of noise in the input, the following test was performed. Originally, five synthetic samples i.e., Trial's 1 to 5, of *noiseless* synthetic muscle sEMG signals are assumed, as shown in Fig. 7. In practical applications, signals obtained from measurement devices such as sEMG sensors contain noise in their content. Therefore, three cases were developed by adding Gaussian noise $(\mathcal{N}(\mu, \sigma))$ with zero mean $(\mu = 0)$ and increasing levels of standard deviations (σ) to the input synthetic sEMG signals as mentioned in Table 2. As the maximum value of the *noiseless* sEMG signals is 1, the chosen σ's were kept within 10% - 20% of the signal maximum for a reasonable level of noise. Restricting it between 10% - 20% is a choice as having higher noise levels (>20%) would dominate over the

underlying 'noiseless' periodic sinusoidal signal, leading to non-physiological synthetic sEMG signals. The corresponding output motions are generated by passing the noisy sEMG as input to the FD equations in Section 2. The following training procedures were performed for each of the three cases.

1-scale Training

The mixed frequency input sEMG signals and corresponding output motion data q at scale [0], denoted by $D^{[0]}$ and $q^{[0]}$, respectively, are mapped to get a baseline performance. This is termed as *1-scale* training as only the full-scale (i.e., [0]) of the mixed frequency data is used for training.

2-scale Training

a. Initiate learning from a *coarse* scale representation of the mixed frequency input data at scale $[-1]$ and map $D^{[-1]}$ to the corresponding motion data at scale [-1], $q^{[-1]}$, of that case.

b. Transfer parameters to the next scale training and finish the learning by mapping $D^{[0]}$ to $q^{[0]}$.

3-scale Training

a. Start learning from a *coarse* scale representation of the mixed frequency input data at scale $[-2]$ and map $D^{[-2]}$ to the corresponding motion data at scale [-2] , $q^{[-2]}$, of that case.

b. Transfer parameters to the next scale training and continue learning by mapping $D^{[-1]}$ to $q^{[-1]}$.

c. Transfer parameters to the next scale training and finish the learning by mapping $D^{[0]}$ to $q^{[0]}$.

Table 2: Input data and gaussian noise level for each case.

Case ID	Input Synthetic sEMG data $+ \mathcal{N}(\mu, \sigma)$
1	Original $+\mathcal{N}(0,0.1)$
2	Original $+\mathcal{N}(0,0.15)$
3	Original $+\mathcal{N}(0,0.2)$

For each case and for each of the training scales in that case, the training data samples contained the data of trial's 1, 2, 4, 5 while trial 3 was used for testing, each trial with $n =$ 500 data points. The MSK parameters $\boldsymbol{\Gamma} = \{\Gamma_l\}_{l=1}^4 = \{f_{0,Bi}^M, l_{0,Bi}^M, f_{0,Tri}^M, l_{0,Tri}^M\}$ were chosen to be identified from the training data using the proposed framework. Due to differences in units and physiological nature of the parameters, the conditioning of the parameter identification system could be affected. To mitigate this issue, normalization [1,44] was applied to each of the parameters,

$$\bar{\Gamma}_l = \frac{\Gamma_l}{\Gamma_l^{(0)}} \tag{37}$$

where $\Gamma_l^{(0)}$ was the initial value of the parameter. Therefore, the parameters to be identified became $\bar{\boldsymbol{\Gamma}} = \{\bar{\Gamma}_l\}_{l=1}^4$.

The proposed framework, as described in Section 3, was applied to each case to simultaneously learn the MSK forward dynamics surrogate and identify the MSK parameters $\bar{\boldsymbol{\Gamma}}$ by optimizing Eq. (16), where the residual of the governing equation for the current time step k, was expressed as

$$
\begin{aligned}
r&\left(\hat{q}_k^{[-j]}(\boldsymbol{\theta_q}), \dot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}), \ddot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}); \boldsymbol{\Gamma}(\bar{\boldsymbol{\Gamma}}; \boldsymbol{\Gamma}^{(0)})\right) \\
&= I\ddot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}) - E\left(\hat{q}_k^{[-j]}(\boldsymbol{\theta_q})\right) \\
&\quad - T^{MT}\left(a_{Bi}(t_k), a_{Tri}(t_k), \hat{q}_k^{[-j]}(\boldsymbol{\theta_q}), \dot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}); \boldsymbol{\Gamma}(\bar{\boldsymbol{\Gamma}}; \boldsymbol{\Gamma}^{(0)})\right)
\end{aligned}
\tag{38}
$$

and is included in the residual term J_{res} in the loss function in Eq. (16). While the training happens sequentially from coarse to fine-scales of the motion, the final identification of parameters happens at the scale [0], i.e., the full-scale in each of the 1-, 2- and 3-scale MR training types.

A GRU with 2 history steps, 1 hidden layer and 50 neurons in each layer was used. The training was performed after standardizing the data, such that the scale or range of the input and output have minimal influence on model performance [21]. The Adam algorithm [65] was used with an initial learning rate of 1×10^{-3} and the penalty parameter for the MSK residual term in the loss function, $\beta \propto \frac{\Delta t^2}{I} = 10^{-3}$. Δt is the time-step between data points and I is the moment of inertia

in Eq. (38). Five parameter initialization seeds were used for an averaged response of the MR training.

To compare the post-training performance of 1-, 2- and 3-scale MR training's, the average testing mean squared error (MSE) and testing R^2 scores were compared, where these measures for a single trial are defined as:

$$\text{MSE} = \frac{1}{n}\|\boldsymbol{q} - \widehat{\boldsymbol{q}}\|_{L_2}^2 \tag{39}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(q_i - \widehat{q}_i)^2}{\sum_{i=1}^{n}(q_i - \bar{q})^2} \tag{40}$$

where $\boldsymbol{q}$ is the motion data of the trial, $\widehat{\boldsymbol{q}}$ is the trial's predicted motion from the MR PI-RNN framework, and $\bar{q}$ is the mean of trial's motion data with n being the number of data points in the trial. At each epoch in the MR training, the *training loss* is calculated by using the scale of the *training data* used in that training scale, i.e., scale $[-j]$ of the data is used in j-scale training.

The gradual improvement in these metrics is evident from Fig. 8 where, as further scales of information are added and the training data is augmented, the generalization performance shows improvement from 1-scale to 3-scale. Overall, it is noted that the test metrics such as the MSE reduces, and the R^2 score gets closer to one, indicating an increase in the generalization accuracy as more training scales are introduced. This can be explained through the theory of bias-variance tradeoff; training on various scales of the data introduces more variance to the training, helping the ML framework to reduce the bias it develops by just training on the full-scale of the data. Together, this reduction in bias and growth in variance leads to a better generalization performance. Computationally, this method improves accuracy in the same amount of training epochs showing the efficiency of this method. As generalization predictions post-training are made using the full-scale of the data, there is no increase in time needed to perform the forward pass for any scale.

Meanwhile, the MSK parameters, f_0^M (maximum isometric force) and l_0^M (optimal muscle length corresponding to the maximum isometric force), of both the biceps and the triceps were accurately identified from the motion data, as shown in Table 3. Compared with the parameter identification

from our previous work [1] where in addition to f_0^M, the maximum contraction velocity v_{max}^M was independently identified, due to non-convergence of l_0^M by the time-domain and feature-encoded trainings, the proposed method can accurately identify l_0^M. v_{max}^M can then by obtain by the experimentally observed relationship of $v_{max}^M/l_0^M = 10s^{-1}$ [57,66].

For the identification of optimal muscle length parameters (l_0^M), the initial points need to be chosen with respect to constraints applied by the geometry of the MSK system. The errors reported in Table 3 are calculated by taking the average of the percentage error of the identified MSK parameters from the 3-scale training with the multiple parameter $(\boldsymbol{\theta}, \boldsymbol{\Gamma})$ initializations. It was observed that in MSK parameter identification, similar accuracy in characterization was obtained from all training scale approaches used within each case, with errors less than 1%. This indicates that the MR PI-RNN improves the generalization performance of the motion prediction, without loss in parameter identification accuracy. It is noted that this example investigates the predictivity of in-distribution testing data, i.e., testing data that lies within the range of the training data. The effect of MR PI-RNN training on out-of-distribution predictivity is also studied in Appendix E.
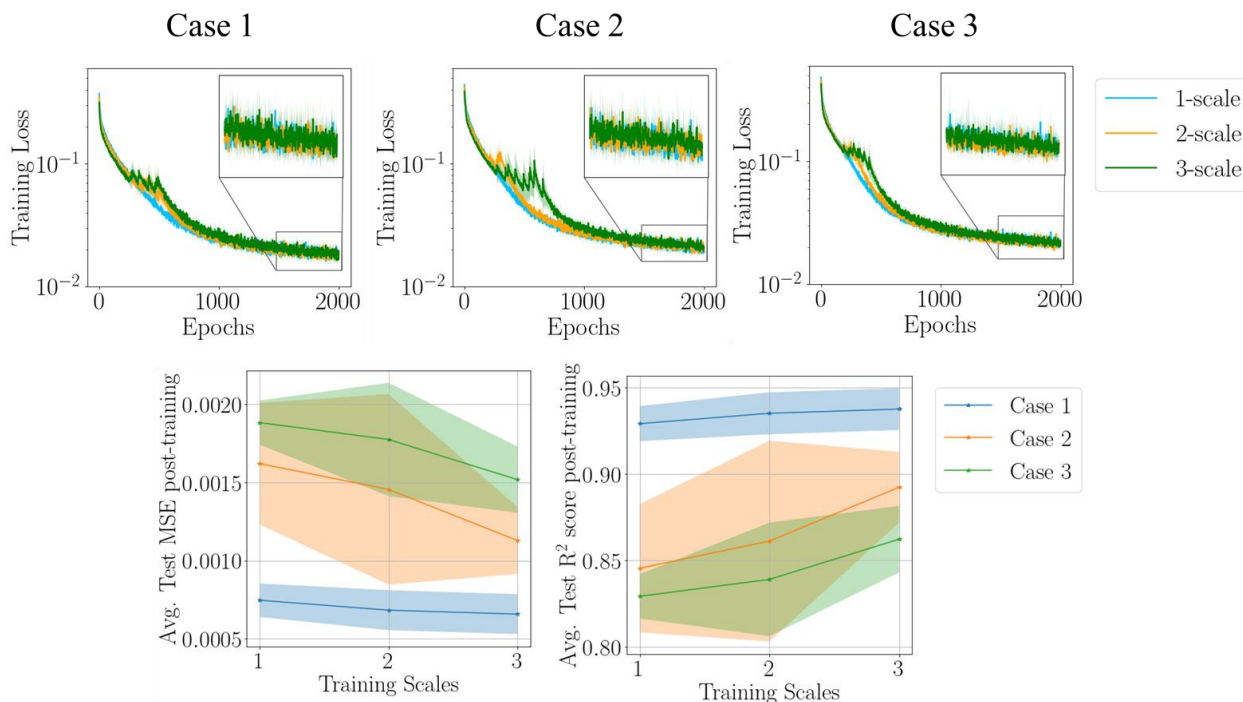
Fig. 8 The training loss and testing metrics are shown. The zoomed-in plots are included for clarity on the loss evolution for the last 500 epochs. The shaded area indicates one standard deviation from the mean (solid line), in both the loss and average test MSE and R^2 score figures. As more

scales of data are introduced in the MR training, the average Test MSE and R^2 score calculated post-training improve in each case.

Table 3: The average percentage error (shown as mean $\pm$ standard deviation) between predicted and true values of the parameters for 3-scale training for each case from five initialization points.

Parameter	Case 1	Case 2	Case 3
$f^M_{0,Bi}$	0.50 ± 0.02	0.50 ± 0.01	0.37 ± 0.04
$f^M_{0,Tri}$	0.06 ± 0.02	-0.04 ± 0.02	-0.02 ± 0.03
$l^M_{0,Bi}$	0.10 ± 0.03	0.10 ± 0.02	0.05 ± 0.05
$l^M_{0,Tri}$	-0.06 ± 0.07	-0.05 ± 0.02	-0.03 ± 0.10

5. Validation: Elbow Flexion-Extension Motion

5.1 Application of MR PI-RNN to Subject-Specific Data

The recorded motion data and sEMG signals were collected and processed as per the data acquisition protocols mentioned in [1]. In brief, three elbow flexion-extension motion trials were performed by the subject for 10 seconds each, with two Delsys Trigno sEMG sensors placed on the biceps and triceps muscle groups, based on SENIAM recommendations [67] . The processed sEMG signals were transformed as described in Section 2.1 to obtain muscle activation signals, used to calculate the MSK forward dynamics ODE residual. The same simplified rigid body model was used as in Section 4 and appropriately scaled anthropometric properties (for the geometry of the model) and physiological parameters (for muscle-tendon material models used for the muscle groups) based on the generic upper body model defined in [68,69] were used. Fig. 9 shows the measured data of the three trials, including the transient raw sEMG signals and the corresponding angular motion of the elbow flexion-extension of the subject.
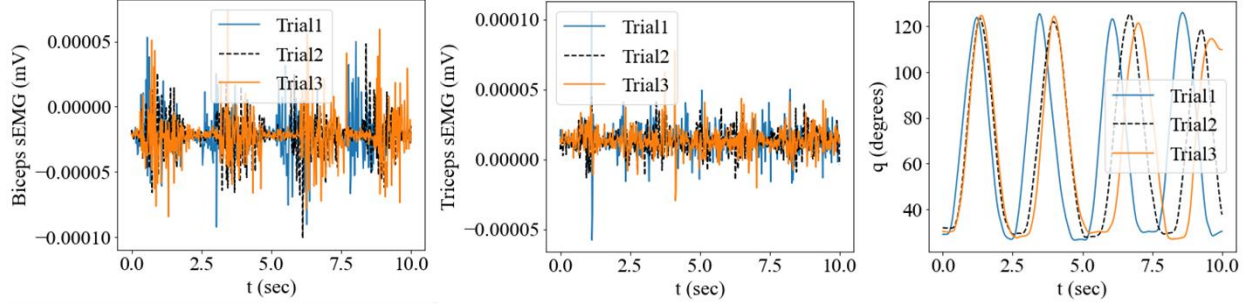
Fig. 9 The measured raw sEMG signals and the corresponding angular motion of the elbow flexion-extension of the subject are plotted.

In this example, the raw sEMG signals were used as input. A 5-scale MR training procedure as described in Section 4 was used on a GRU with 1 hidden layer with 50 neurons. The data of trials 1 and 3 were used for training, while trial 2 was used for testing, where each signal contained 500 temporal data points.

The muscle parameters to be identified by the framework include the maximum isometric force and the optimal muscle length from both muscle groups, which are denoted as $\boldsymbol{\Gamma} = \{f_{0,Bi}^M, l_{0,Bi}^M, f_{0,Tri}^M, l_{0,Tri}^M\}$. It was observed in our tests that despite the normalization process described in Eq. (37) and (38), the parameters obtained at the end of the MR training with motion data either diverged or converged to non-physiological values. To obtain physiologically consistent parameters, we use the values obtained from literature studies and constrain the space of parameter search [44].

Let the parameter to be identified be defined as

$$\Gamma_l(\boldsymbol{\psi}) = \frac{1}{N} \sum_{r=1}^{N} \bar{\gamma}_r \text{sig}(\psi_r), \boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_N] \tag{41}$$

where $\bar{\gamma}_r$ is the value defined in the rth literature study and ψ_r is the parameter to be optimized in the training such that it can be used to evaluate the sigmoid function $\text{sig}(\psi_r)$ and $\boldsymbol{\psi}$ is the vector of these trainable parameters. Using the optimized $\boldsymbol{\psi}$, the desired MSK parameters can be estimated. This formulation constrains the identified parameters to be consistent with parameters obtained through experimental studies [68–70].

The proposed framework was then applied to simultaneously learn the MSK forward dynamics surrogate and identify the MSK parameters $\boldsymbol{\Gamma}$ by optimizing Eq. (16), where the residual of the governing equation for the current time step k, is expressed with a slight correction of Eq. (38) due to parameter space modification as

$$
\begin{aligned}
& r\left(\hat{q}_k^{[-j]}(\boldsymbol{\theta_q}), \dot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}), \ddot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}); \boldsymbol{\Gamma}(\boldsymbol{\psi})\right) \\
& = I\ddot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}) - E\left(\hat{q}_k^{[-j]}(\boldsymbol{\theta_q})\right) - T^{MT}\left(a_{Bi}(t_k), a_{Tri}(t_k), \hat{q}_k^{[-j]}(\boldsymbol{\theta_q}), \dot{\hat{q}}_k^{[-j]}(\boldsymbol{\theta_q}); \boldsymbol{\Gamma}(\boldsymbol{\psi})\right).
\end{aligned}
\tag{42}
$$

This is introduced into the residual term J_{res} in the loss function and the optimization problem becomes,

$$
\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\psi}} = \underset{\boldsymbol{\theta},\boldsymbol{\psi}}{\operatorname{argmin}}\left(J_{data}(\boldsymbol{\theta}) + \beta J_{res}(\boldsymbol{\theta}, \boldsymbol{\psi})\right).
\tag{43}
$$

As mentioned in the verification example (Section 4), the multi-resolution parameter identification is performed starting from the coarsest scale, transferring the learned hyperparameters to the next finer scale parameter identification, and finally completing the parameter identification at the full-scale, i.e., at scale [0].

5.2 Results

To accelerate the training process, the training dataset is standardized to have zero mean and unit variance. The training was performed with the standardized data, using the Adam algorithm [65] with an initial learning rate of 1×10^{-3} and 4 history steps were considered. Five parameter initialization seeds were used for an averaged response of the MR training. To quantify the error in the testing predictions, a normalized mean squared error (NMSE) was defined,

$$
\text{NMSE} = \frac{1}{n}\frac{\sum_{i=1}^{n}(q_i - \hat{q}_i)^2}{\sum_{i=1}^{n}(q_i - \bar{q})^2}
\tag{44}
$$

where q_i is the ith target motion data point, $\hat{q}_i$ is the ith predicted motion data point, from the MR PI-RNN framework, and $\bar{q}$ is the mean of target motion data. The R^2 score was calculated using the metric defined in Eq. (40). From Fig. 10, Fig. 11 and Table 4, it is clear that addition of training

29

scales leads to improved motion predictions. The multi-resolution training leads to an increase in average test R^2 score of more than 40% (bringing it closer to one), averaged over the multiple initialization seeds. With the addition of more scales, Fig. 10 clearly shows the progression in improvement of the predictions as more scales are involved in the training.

| 1-scale | 2-scale | 3-scale | 4-scale |
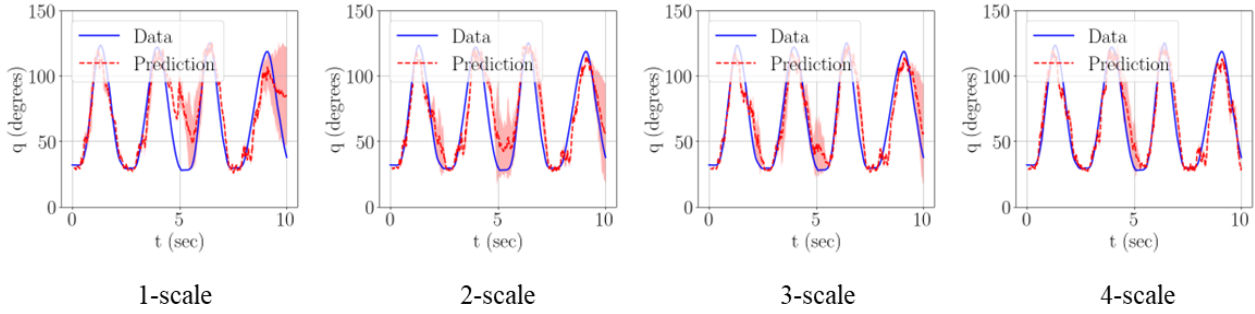
Fig. 10 Comparison of test predictions post-training for each MR training scale performed. The solid dash line is the mean of the predictions post-training when various initialization points are utilized to begin the MR training, with shaded region indicating one standard deviation from the mean.
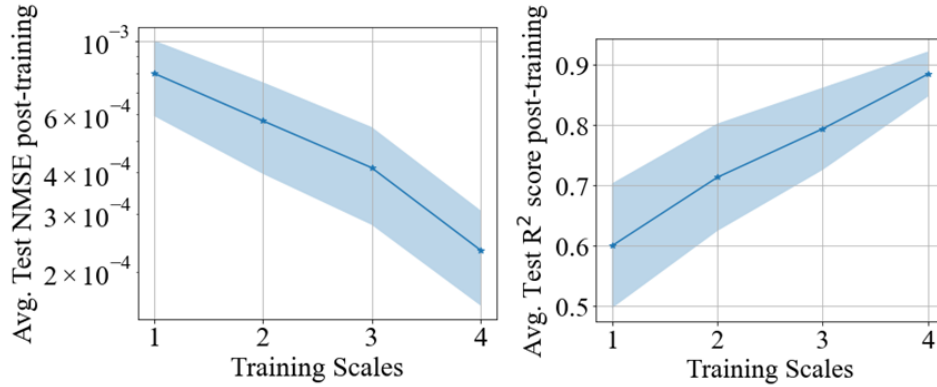
Fig. 11 The test normalized mean squared error (NMSE) and test R^2 score are plot for the testing predictions post-training, averaged over five initialization seeds. The mean of the metric is the solid marker line, the shaded portion being one standard deviation from the mean.

Table 4: The test metrics such as NMSE and R^2 score averaged over five initialization seeds, for the various training scales involved are reported here. The % decrease in average NMSE and % increase in average R^2 w.r.t 1-scale training are shown in the 3rd and 5th columns respectively.

Training Scale	Avg. Test NMSE	Decrease (%) in Avg. Test NMSE w.r.t 1-scale Training	Avg. Test R^2 Score	Increase (%) in Avg. Test R^2 score w.r.t 1-Scale training
1	8.00E-04	-	0.599	-
2	5.74E-04	28%	0.713	19%
3	4.14E-04	48%	0.793	32%
4	2.33E-04	71%	0.884	47%

The identified MSK parameters from the MR PI-RNN training are summarized in Table 5 with the mean of the final converged values of f_0^M and l_0^M obtained from multiple parameter initializations at 4-scale training, consistent with the physiological estimates of these parameters reported in literature [68–70]. $l_{0,Bi}^M$ is slightly outside the estimated range, which could be attributed to the variance in population. Similar values were obtained across all scales of training hence parameters obtained from a representative 4-scale training are shown here. The results demonstrated the effectiveness of the proposed MR PI-RNN framework and promising potential for real applications.

Table 5: The identified parameter estimates using MR PI-RNN training, and their values reported in literature [68–70].

Parameter	Identified values	Estimates from literature
$f_{0,Bi}^M$ (N)	348.23 ± 0.2	158.4-845
$l_{0,Bi}^M$ (m)	0.108 ± 0.001	0.115-0.142
$f_{0,Tri}^M$ (N)	758.36 ± 0.5	554.4-2332.916
$l_{0,Tri}^M$ (m)	0.069 ± 0.001	0.067-0.087

6. Discussion and Conclusions

In this work, we proposed a multi-resolution physics-informed recurrent neural network (MR PI-RNN) for an application to MSK systems, for time-domain motion prediction and parameter identification. A GRU with a physics-informed loss function that minimized the error in the training data and the residual of the MSK forward dynamics equilibrium was used for this purpose. Wavelet based multi-resolution techniques were used to decompose the input sEMG signals and output joint motion data into coarse-scale approximations at different scales and fine-scale details at those scales. The sEMG and joint motion multi-scale components were then mapped to each other starting from a chosen coarse-scale components and then sequentially trained (via transfer learning) to higher scales, completing the training on the full-scale of the data.

By initializing training on the coarse-scale of the training data, the optimization reaches a local minimum that serves as a better initialization state for the training data that includes the sequential fine-scale details. The proposed transfer-learning based sequential training scheme can be used for learning datasets that have high frequency signals as shown in the verification example with synthetic mixed frequency sEMG data. The numerical examples show an improvement in testing prediction and identifying the parameters. We observe from the loss profiles that the testing loss decreased while the training loss increased as more scales of data were brought in. It was also observed that the average test MSE and R^2 metrics showed a clear improvement in the generalization accuracy. These phenomena can be explained through the theory of bias-variance tradeoff; training on various scales of the data introduces more variance to the training, helping the ML framework to reduce the bias it develops by just training on the full-scale of the data. Computationally, it is noted that the proposed method achieves improved accuracy by using the same amount of training epochs.

The proposed MR framework was validated on recorded sEMG and motion data from a subject [1] and significant improvements were observed in the testing prediction accuracy, with 1-scale training often leading to large errors. The predicted motion at higher training scales showed improvements across all initialization points used, indicating the robustness of the method. The identified parameters were also consistent with the physiological range observed in literature.

This method also has the advantage of operating in the time-domain as compared to the feature-encoded (FE) training [1], where the input sEMG signals were projected on to the frequency domain using the Fourier basis. In the FE training, to make a prediction, the input signal for the entire duration of the movement prediction was needed whereas the physics informed MR training of the RNN enables the trained model to make real-time predictions by using the information of the previous time-steps and the current sEMG signal. In addition, for mixed frequency signals, wavelet resolution can better capture the local frequency information as compared to the Fourier basis which captures the global frequency information. As compared to the NN-based time-domain training performed at the original scale of the data (scale [0]) proposed in [1], the MR PI-RNN training approach described here achieved significant improvements due to the stronger sequence learning capability of the RNN and the ability of the MR training.

This method is presented as a general approach where multi-resolution is applied to both input and output. For some applications, e.g., those that require only data mapping, the MR training can be applied by only considering the decomposition to the input, keeping the output at the full-scale (i.e., scale [0]) throughout, or vice versa. To apply this method to clinical studies, RNN hyperparameters may also be tuned to account for subject variability. The dependence of this method on the number of data points available in a signal can also be studied in the future. To further improve this method, we can consider the use of multi-resolution as activation functions of the ML framework, instead of relying on data filtration processes for better computational efficiency. This method will also be studied on other physics-informed ML techniques to solve forward problems with PDEs having mixed frequency source terms.

Acknowledgment

Nomenclature

e Raw sEMG signals captured by sensors

u	Neural excitations
a	Muscle activations
d	Electro-mechanical delay between origin of neural excitation from central nervous system and reaching the muscle group
A	Shape factor used to relate neural excitation to muscle activation
f_0^M	Maximum isometric force in the muscle
l_0^M	Optimal muscle length corresponding to the maximum isometric force
v_{max}^M	Maximum contraction velocity
l_s^T	Slack length of the tendon
ϑ	Pennation angle between muscle and tendon
$\boldsymbol{\kappa}$	Set of muscle parameters for each muscle group
$\tilde{l}^M$	Normalized muscle length
$\tilde{v}^M$	Normalized muscle velocity
l^{MT}	Total length of muscle-tendon complex
f^A	Active force component of hill-type muscle model
$f^{A,L}$	Length dependent active force generation component
f^V	Velocity dependent active force generation component
f^P	Passive force component of hill-type muscle model
F^M	Total muscle force

F^{MT}	Total force produced by muscle-tendon complex
T^{MT}	Torque produced by the muscle-tendon complex
q	Generalized angular motion of the MSK system
h_n	Hidden state after n history steps of the RNN/GRU
W_{XY}	Weights connecting the variable X to variable Y
b_X	Bias for the RNN to calculate variable X
r_i	Output from the reset gate of the GRU
u_i	Output from the update gate of the GRU
$z_i, c_i, \tilde{c}_i$	Intermediate variables in GRU forward pass
$\tilde{h}_n$	Candidate hidden state after n history steps of the GRU
θ	Set of all weights and biases of the RNN/GRU
$\phi_{j,k}$	Scaling function in multi-resolution analysis
$\psi_{j,k}$	Wavelet function in multi-resolution analysis
V_j, W_j	The nested and complementary subspaces containing $\phi_{j,k}$ and $\psi_{j,k}$ respectively
P_j, H_j	Projection operators on a function f projecting it onto subspaces V_j and W_j respectively.
Γ_i	i^{th} parameter characterizing the parameterized ODE system
$x_i^{[-j]}$	Input data at scale $[-j]$ to the MR PI-RNN framework at time-step i
$D^{[-j]}$	Input data set projected to scale $[-j]$

$\hat{q}_i^{[-j]}$	Predicted angular motion at scale [-j] from the MR PI-RNN framework at time-step i
J	Composite loss function of the PI-PINN minimizing data and ODE residual
l_{fa}	Forearm length
l_{ua}	Upper arm length
m_{fa}	Mass of forearm

References

[1] Taneja, K., He, X., He, Q., Zhao, X., Lin, Y.-A., Loh, K. J., and Chen, J.-S., 2022, "A Feature-Encoded Physics-Informed Parameter Identification Neural Network for Musculoskeletal Systems," J. Biomech. Eng., **144**(121006), pp. 1–16.

[2] Costello, Z., and Martin, H. G., 2018, "A Machine Learning Approach to Predict Metabolic Pathway Dynamics from Time-Series Multiomics Data," npj Syst Biol Appl, **4**(1), pp. 1–14.

[3] Damle, C., and Yalcin, A., 2007, "Flood Prediction Using Time Series Data Mining," Journal of Hydrology, **333**(2), pp. 305–316.

[4] Nishino, T., and Hokugo, A., 2020, "A Stochastic Model for Time Series Prediction of the Number of Post-Earthquake Fire Ignitions in Buildings Based on the Ignition Record for the 2011 Tohoku Earthquake," Earthquake Spectra, **36**(1), pp. 232–249.

[5] Manevitz, L., Bitar, A., and Givoli, D., 2005, "Neural Network Time Series Forecasting of Finite-Element Mesh Adaptation," Neurocomputing, **63**, pp. 447–463.

[6] Frank, R. J., Davey, N., and Hunt, S. P., "Time Series Prediction and Neural Networks."

[7] Längkvist, M., Karlsson, L., and Loutfi, A., 2014, "A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling," Pattern Recognition Letters, **42**, pp. 11–24.

[8] Calderón-Macías, C., Sen, M. K., and Stoffa, P. L., 2000, "Artificial Neural Networks for Parameter Estimation in Geophysics," Geophysical Prospecting, **48**(1), pp. 21–47.

[9] Hubbard, S. S., and Rubin, Y., 2000, "Hydrogeological Parameter Estimation Using Geophysical Data: A Review of Selected Techniques," Journal of Contaminant Hydrology, **45**(1), pp. 3–34.

[10] Ebrahimian, H., Astroza, R., and Conte, J. P., 2015, "Extended Kalman Filter for Material Parameter Estimation in Nonlinear Structural Finite Element Models Using Direct Differentiation Method," Earthquake Engineering & Structural Dynamics, **44**(10), pp. 1495–1522.

[11] Haghighat, E., Raissi, M., Moure, A., Gomez, H., and Juanes, R., 2021, "A Physics-Informed Deep Learning Framework for Inversion and Surrogate Modeling in Solid Mechanics," Comput. Methods Appl. Mech. Eng., **379**, p. 113741.

[12] Heine, C. B., and Menegaldo, L. L., 2018, "Numerical Validation of a Subject-Specific Parameter Identification Approach of a Quadriceps Femoris EMG-Driven Model," Medical Engineering & Physics, **53**, pp. 66–74.

[13] Hirschvogel, M., Bassilious, M., Jagschies, L., Wildhirt, S. M., and Gee, M. W., 2017, "A Monolithic 3D-0D Coupled Closed-Loop Model of the Heart and the Vascular System: Experiment-Based Parameter Estimation for Patient-Specific Cardiac Mechanics," International Journal for Numerical Methods in Biomedical Engineering, **33**(8), p. e2842.

[14] Schmid, H., Nash, M. P., Young, A. A., and Hunter, P. J., 2006, "Myocardial Material Parameter Estimation—A Comparative Study for Simple Shear," Journal of Biomechanical Engineering, **128**(5), pp. 742–750.

[15] Pau, J. W. L., Xie, S. S. Q., and Pullan, A. J., 2012, "Neuromuscular Interfacing: Establishing an EMG-Driven Model for the Human Elbow Joint," IEEE Trans. Biomed. Eng., **59**(9), pp. 2586–2593.

[16] Zhao, Y., Zhang, Z., Li, Z., Yang, Z., Dehghani-Sanij, A. A., and Xie, S., 2020, "An EMG-Driven Musculoskeletal Model for Estimating Continuous Wrist Motion," IEEE Trans. Neural Syst. Rehabil. Eng., **28**(12), pp. 3113–3120.

[17] Kian, A., Pizzolato, C., Halaki, M., Ginn, K., Lloyd, D., Reed, D., and Ackland, D., 2019, "Static Optimization Underestimates Antagonist Muscle Activity at the Glenohumeral Joint: A Musculoskeletal Modeling Study," J. Biomech., **97**, p. 109348.

[18] Buchanan, T. S., Lloyd, D. G., Manal, K., and Besier, T. F., 2004, "Neuromusculoskeletal Modeling: Estimation of Muscle Forces and Joint Moments and Movements from Measurements of Neural Command," J. Appl. Biomech., **20**(4), pp. 367–395.

[19] Lloyd, D. G., and Besier, T. F., 2003, "An EMG-Driven Musculoskeletal Model to Estimate Muscle Forces and Knee Joint Moments in Vivo," J. Biomech., **36**(6), pp. 765–776.

[20] Buongiorno, D., Barsotti, M., Barone, F., Bevilacqua, V., and Frisoli, A., 2018, "A Linear Approach to Optimize an EMG-Driven Neuromusculoskeletal Model for Movement Intention Detection in Myo-Control: A Case Study on Shoulder and Elbow Joints," Front. Neurorobot., **12**, p. 74.

[21] Goodfellow, I., Bengio, Y., and Courville, A., 2016, *Deep Learning*, MIT Press, Cambridge, MA, USA.

[22] Kaneko, S., Wei, H., He, Q., Chen, J.-S., and Yoshimura, S., 2021, "A Hyper-Reduction Computational Method for Accelerated Modeling of Thermal Cycling-Induced Plastic Deformations," J. Mech. Phys. Solids, **151**, p. 104385.

[23] Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B., 2019, "Deep Fluids: A Generative Network for Parameterized Fluid Simulations," Comput. Graph. Forum, **38**(2), pp. 59–70.

[24] Xie, X., Zhang, G., and Webster, C., 2019, "Non-Intrusive Inference Reduced Order Model for Fluids Using Deep Multistep Neural Network," Mathematics, **7**(8), p. 757.

[25] Fries, W. D., He, X., and Choi, Y., 2022, "LaSDI: Parametric Latent Space Dynamics Identification," Comput. Methods Appl. Mech. Eng., **399**, p. 115436.

[26] He, X., Choi, Y., Fries, W. D., Belof, J. L., and Chen, J.-S., 2023, "GLaSDI: Parametric Physics-Informed Greedy Latent Space Dynamics Identification," J. Comput. Phys., **489**, p. 112267.

[27] Ghaboussi, J., Garrett, J. H., and Wu, X., 1991, "Knowledge-Based Modeling of Material Behavior with Neural Networks," J. Eng. Mech. - ASCE, **117**(1), pp. 132–153.

[28] Lefik, M., Boso, D. P., and Schrefler, B. A., 2009, "Artificial Neural Networks in Numerical Modelling of Composites," Comput. Methods Appl. Mech. Eng., **198**(21), pp. 1785–1804.

[29] He, X., and Chen, J.-S., 2022, "Thermodynamically Consistent Machine-Learned Internal State Variable Approach for Data-Driven Modeling of Path-Dependent Materials," Comput. Methods Appl. Mech. Eng., **402**, p. 115348.

[30] Kirchdoerfer, T., and Ortiz, M., 2017, "Data Driven Computing with Noisy Material Data Sets," Comput. Methods Appl. Mech. Eng., **326**, pp. 622–641.

[31] Eggersmann, R., Kirchdoerfer, T., Reese, S., Stainier, L., and Ortiz, M., 2019, "Model-Free Data-Driven Inelasticity," Comput. Methods Appl. Mech. Eng., **350**, pp. 81–99.

[32] He, Q., and Chen, J.-S., 2020, "A Physics-Constrained Data-Driven Approach Based on Locally Convex Reconstruction for Noisy Database," Comput. Methods Appl. Mech. Eng., **363**, p. 112791.

[33] He, X., He, Q., Chen, J.-S., Sinha, U., and Sinha, S., 2020, "Physics-Constrained Local Convexity Data-Driven Modeling of Anisotropic Nonlinear Elastic Solids," DCE, **1**, p. e19.

[34] He, Q., Laurence, D. W., Lee, C.-H., and Chen, J.-S., 2021, "Manifold Learning Based Data-Driven Modeling for Soft Biological Tissues," J. Biomech., **117**, p. 110124.

[35] Bahmani, B., and Sun, W., 2022, "Manifold Embedding Data-Driven Mechanics," J. Mech. Phys. Solids, **166**, p. 104927.

[36] Carrara, P., De Lorenzis, L., Stainier, L., and Ortiz, M., 2020, "Data-Driven Fracture Mechanics," Comput. Methods Appl. Mech. Eng., **372**, p. 113390.

[37] He, X., He, Q., and Chen, J.-S., 2021, "Deep Autoencoders for Physics-Constrained Data-Driven Nonlinear Materials Modeling," Comput. Methods Appl. Mech. Eng., **385**, p. 114034.

[38] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2019, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," J. Comput. Phys., **378**, pp. 686–707.

[39] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., 2021, "Physics-Informed Machine Learning," Nat. Rev. Phys., **3**(6), pp. 422–440.

[40] He, Q., and Tartakovsky, A. M., 2021, "Physics-Informed Neural Network Method for Forward and Backward Advection-Dispersion Equations," Water Resour. Res., **57**(7).

[41] Zhu, Q., Liu, Z., and Yan, J., 2021, "Machine Learning for Metal Additive Manufacturing: Predicting Temperature and Melt Pool Fluid Dynamics Using Physics-Informed Neural Networks," Comput. Mech., **67**(2), pp. 619–635.

[42] Xu, K., Tartakovsky, A. M., Burghardt, J., and Darve, E., 2021, "Learning Viscoelasticity Models from Indirect Data Using Deep Neural Networks," Comput. Methods Appl. Mech. Eng., **387**, p. 114124.

[43] Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., and Barajas-Solano, D., 2020, "Physics-Informed Deep Neural Networks for Learning Parameters and Constitutive Relationships in Subsurface Flow Problems," Water Resour. Res., **56**(5), p. e2019WR026731.

[44] He, Q., Stinis, P., and Tartakovsky, A. M., 2022, "Physics-Constrained Deep Neural Network Method for Estimating Parameters in a Redox Flow Battery," J. Power Sources, **528**, p. 231147.

[45] He, Q., Barajas-Solano, D., Tartakovsky, G., and Tartakovsky, A. M., 2020, "Physics-Informed Neural Networks for Multiphysics Data Assimilation with Application to Subsurface Transport," Adv. Water Resour., **141**, p. 103610.

[46] Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., and Perdikaris, P., 2020, "Machine Learning in Cardiovascular Flows Modeling: Predicting Arterial Blood Pressure from Non-Invasive 4D Flow MRI Data Using Physics-Informed Neural Networks," Comput. Methods Appl. Mech. Eng., **358**, p. 112623.

[47] Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D. E., and Kuhl, E., 2020, "Physics-Informed Neural Networks for Cardiac Activation Mapping," Front. Phys., **8**.

[48] Alber, M., Buganza Tepole, A., Cannon, W. R., De, S., Dura-Bernal, S., Garikipati, K., Karniadakis, G., Lytton, W. W., Perdikaris, P., Petzold, L., and Kuhl, E., 2019, "Integrating Machine Learning and Multiscale Modeling—Perspectives, Challenges, and Opportunities in the Biological, Biomedical, and Behavioral Sciences," npj Digit. Med., **2**(1), pp. 1–11.

[49] Yazdani, A., Lu, L., Raissi, M., and Karniadakis, G. E., 2020, "Systems Biology Informed Deep Learning for Inferring Parameters and Hidden Dynamics," PLOS Comput. Biol., **16**(11), p. e1007575.

[50] Zhang, J., Zhao, Y., Bao, T., Li, Z., Qian, K., Frangi, A. F., Xie, S. Q., and Zhang, Z.-Q., 2023, "Boosting Personalized Musculoskeletal Modeling With Physics-Informed Knowledge Transfer," IEEE Trans. Instrum. Meas., **72**, pp. 1–11.

[51] Bartsoen, L., Faes, M. G. R., Andersen, M. S., Wirix-Speetjens, R., Moens, D., Jonkers, I., and Sloten, J. V., 2023, "Bayesian Parameter Estimation of Ligament Properties Based on Tibio-Femoral Kinematics during Squatting," Mech. Syst. Signal Process., **182**, p. 109525.

[52] Linden, N. J., Kramer, B., and Rangamani, P., 2022, "Bayesian Parameter Estimation for Dynamical Models in Systems Biology," PLoS Comput. Biol., **18**(10), p. e1010651.

[53] Lipton, Z. C., Berkowitz, J., and Elkan, C., 2015, "A Critical Review of Recurrent Neural Networks for Sequence Learning," arXiv:1506.00019 [cs].

[54] Hamner, S. R., Seth, A., and Delp, S. L., 2010, "Muscle Contributions to Propulsion and Support during Running," J Biomech, **43**(14), pp. 2709–2716.

[55] Zhang, Y., Chen, J., He, Q., He, X., Basava, R. R., Hodgson, J., Sinha, U., and Sinha, S., 2020, "Microstructural Analysis of Skeletal Muscle Force Generation during Aging," Int. J. Numer. Methods Biomed. Eng., **36**(1).

[56] Chen, J.-S., Basava, R. R., Zhang, Y., Csapo, R., Malis, V., Sinha, U., Hodgson, J., and Sinha, S., 2016, "Pixel-Based Meshfree Modelling of Skeletal Muscles," Comput. Methods Biomech. Biomed. Eng. Imaging Vis., **4**(2), pp. 73–85.

[57] Millard, M., Uchida, T., Seth, A., and Delp, S. L., 2013, "Flexing Computational Muscle: Modeling and Simulation of Musculotendon Dynamics," J. Biomech. Eng., **135**(2), p. 021005.

[58] Winters, J. M., 1990, "Hill-Based Muscle Models: A Systems Engineering Perspective," *Multiple Muscle Systems*, Springer New York, New York, NY, pp. 69–93.

[59] Mallat, S. G., 1989, "Multiresolution Approximations and Wavelet Orthonormal Bases of L 2 (R)," Trans. Am. Math. Soc., **315**(1), p. 69.

[60] Mallat, S. G., 1989, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Trans. Pattern Anal. Mach. Intell., **11**(7), pp. 674–693.

[61] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M., 2018, "Automatic Differentiation in Machine Learning: A Survey," J. Mach. Learn. Res., **18**, p. 43.

[62] Yao, Y., Rosasco, L., and Caponnetto, A., 2007, "On Early Stopping in Gradient Descent Learning," Constr Approx, **26**(2), pp. 289–315.

[63] Weiss, K., Khoshgoftaar, T. M., and Wang, D., 2016, "A Survey of Transfer Learning," J. Big Data, **3**(1), p. 9.

[64] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., and van Mulbregt, P., 2020, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," Nat Methods, **17**(3), pp. 261–272.

[65] Kingma, D. P., and Ba, J., 2017, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs].

[66] Thelen, D. G., 2003, "Adjustment of Muscle Mechanics Model Parameters to Simulate Dynamic Contractions in Older Adults," J. Biomech. Eng., **125**(1), pp. 70–77.

[67] Hermens, H. J., Freriks, B., Merletti, R., Stegeman, D., Blok, J., Rau, G., Disselhorst-Klug, C., and Hägg, G., 1999, "European Recommendations for Surface ElectroMyoGraphy."

[68] Garner, B. A., and Pandy, M. G., 2003, "Estimation of Musculotendon Properties in the Human Upper Limb," Ann. Biomed. Eng., **31**(2), pp. 207–220.

[69] Holzbaur, K. R. S., Murray, W. M., and Delp, S. L., 2005, "A Model of the Upper Extremity for Simulating Musculoskeletal Surgery and Analyzing Neuromuscular Control," Ann. Biomed. Eng., **33**(6), pp. 829–840.

[70] An, K. N., Hui, F. C., Morrey, B. F., Linscheid, R. L., and Chao, E. Y., 1981, "Muscles across the Elbow Joint: A Biomechanical Analysis," J. Biomech., **14**(10), pp. 659–669.

Appendix A: Muscle-Tendon Force Generation

The total muscle force F^M can be expressed as

$$F^M\big(a, \tilde{l}^M, \tilde{v}^M; \boldsymbol{\kappa}\big) = f_0^M\left(f^A\big(a, \tilde{l}^M, \tilde{v}^M; \boldsymbol{\kappa}\big) + f^P\big(\tilde{l}^M; \boldsymbol{\kappa}\big)\right), \tag{45}$$

where $f^P\big(\tilde{l}^M\big)$ is the passive muscle length dependent force generation function. The active force f^A component can be expressed as:

$$f^A\big(a, \tilde{l}^M, \tilde{v}^M; \boldsymbol{\kappa}\big) = a f^{A,L}\big(\tilde{l}^M; \boldsymbol{\kappa}\big) f^V\big(\tilde{v}^M; \boldsymbol{\kappa}\big), \tag{46}$$

$$\tilde{l}^M = l^M / l_0^M,$$

$$\tilde{v}^M = v^M / v_{max}^M,$$

where a is the activation function in Eq. (2), $\tilde{l}^M$ is the normalized muscle length, $\tilde{v}^M$ is the normalized velocity of the muscle. The total length of the MT system l^{MT} is given by,

$$l^{MT} = l^M \cos \phi + l^T. \tag{47}$$

Given the current joint angle q and the angular velocity $\dot{q}$, the current length, l^{MT} of the MT system can be calculated using trigonometric relations.

The $f^{A,L}(\tilde{l}^M)$ and $f^V(\tilde{v}^M)$ are generic functions of the length and velocity dependent force generation properties of the active muscle, represented by dimensionless quantities. In this study, the tendon is assumed to be rigid ($l^T = l_s^T$). The total force produced by the MT complex, F^{MT}, can be expressed as:

$$F^{MT}(a, \tilde{l}^M, \tilde{v}^M, \phi; \boldsymbol{\kappa}) = F^M(a, \tilde{l}^M, \tilde{v}^M; \boldsymbol{\kappa}) \cos \phi. \tag{48}$$

The rigid-tendon model simplifies the MT contraction dynamics [57,58] which accounts for the interaction of the activation, force length, and force velocity properties of the MT complex.

Appendix B. Hill-Type Muscle Models

For the length dependent muscle force relations, this work uses the equations given in [55]. The active muscle force dependent on variation in length is given as

$$f^{A,L}(\tilde{l}^M) = \begin{cases} 9(\tilde{l}^M - 0.4)^2, & \tilde{l}^M \le 0.6 \\ 1 - 4(1 - \tilde{l}^M)^2, & 0.6 \le \tilde{l}^M \le 1.4 \\ 9(\tilde{l}^M - 1.6)^2, & \tilde{l}^M > 1.4 \end{cases} \tag{49}$$

$$f^P(\tilde{l}^M) \tag{50}$$

$$= \begin{cases} 0, & \tilde{l}^M \le 1 \\ \gamma_1(\exp(\gamma_2(\tilde{l}^M - 1)) - 1), & 1 \le \tilde{l}^M \le 1.4 \\ (\gamma_1\gamma_2 \exp(0.4\gamma_2))\tilde{l}^M + \gamma_1((1 - 1.4\gamma_2) \exp(0.4\gamma_2) - 1), & \tilde{l}^M > 1.4 \end{cases}$$

Where $\gamma_1 = 0.075$ and $\gamma_2 = 6.6$ correspond to parameters in the passive muscle force model related to an adult human. The muscle force velocity relationship $f^V(\tilde{v}^M)$ is used directly from [66].

Appendix C: Multi-Resolution GRU Formulation

For multi-resolution GRU, the initial learning starts from the coarsest scale $[-j]$ as follows with notations according to Section 3.3.2,

$$
\begin{aligned}
r_i^{[-j]} &= a_\sigma \left(W_{hr}^{[-j]} h_{i-1}^{[-j]} + W_{xr}^{[-j]} x_i^{[-j]} + W_{qr}^{[-j]} q_i^{[-j]} + b_r^{[-j]} \right) \\
u_i^{[-j]} &= a_\sigma \left(W_{hu}^{[-j]} h_{i-1}^{[-j]} + W_{xu}^{[-j]} x_i^{[-j]} + W_{qu}^{[-j]} q_i^{[-j]} + b_u^{[-j]} \right) \\
\tilde{h}_i^{[-j]} &= a_{tanh} \left(r_i^{[-j]} \odot W_{h\tilde{h}}^{[-j]} h_{i-1}^{[-j]} + W_{x\tilde{h}}^{[-j]} x_i^{[-j]} + W_{q\tilde{h}}^{[-j]} q_i^{[-j]} + b_{\tilde{h}}^{[-j]} \right) \\
h_i^{[-j]} &= u_i^{[-j]} \odot h_{i-1}^{[-j]} + \left(1 - u_i^{[-j]} \right) \odot \tilde{h}_i^{[-j]} + b_h^{[-j]}
\end{aligned}
\tag{51}
$$

$$
\forall\, i = n - m, \dots, n - 1,
$$

$$
\begin{aligned}
r_n^{[-j]} &= a_\sigma \left(W_{hr}^{[-j]} h_{n-1}^{[-j]} + W_{xr}^{[-j]} x_n^{[-j]} + b_r^{[-j]} \right) \\
u_n^{[-j]} &= a_\sigma \left(W_{hu}^{[-j]} h_{n-1}^{[-j]} + W_{xu}^{[-j]} x_n^{[-j]} + b_u^{[-j]} \right) \\
\tilde{h}_n^{[-j]} &= a_{tanh} \left(r_n^{[-j]} \odot W_{h\tilde{h}}^{[-j]} h_{n-1}^{[-j]} + W_{x\tilde{h}}^{[-j]} x_n^{[-j]} + b_{\tilde{h}}^{[-j]} \right) \\
h_n^{[-j]} &= u_n^{[-j]} \odot h_{n-1}^{[-j]} + \left(1 - u_n^{[-j]} \right) \odot \tilde{h}_n^{[-j]} + b_h^{[-j]}
\end{aligned}
\tag{52}
$$

$$
\hat{q}_n^{[-j]} = W_{h\hat{q}}^{[-j]} h_n^{[-j]} + b_q^{[-j]}
\tag{53}
$$

where the weights $W_{hr}^{[-j]}, W_{xr}^{[-j]}, W_{qr}^{[-j]}, W_{hu}^{[-j]}, W_{xu}^{[-j]}, W_{qu}^{[-j]}, W_{h\tilde{h}}^{[-j]}, W_{x\tilde{h}}^{[-j]}, W_{q\tilde{h}}^{[-j]}$ and biases $b_r^{[-j]}, b_u^{[-j]}, b_{\tilde{h}}^{[-j]}, b_h^{[-j]}, b_q^{[-j]}$ are trainable parameters.

Appendix D: Equation of Motion of the Simplified MSK Model

The equation of motion for the rigid body system used in Section 4 is,

$$
I\ddot{q} = E(q) + T^{MT}(a_{Bi}, a_{Tri}, q, \dot{q}; \kappa_{Bi}, \kappa_{Tri})
\tag{54}
$$

where,

$$
I = m_{fa} l_{fa}^2
\tag{55}
$$

$$
E(q) = -m_{fa} g l_{fa} \sin(q)
\tag{56}
$$

$$T^{MT}(a_{Bi}, a_{Tri}, q, \dot{q}; \boldsymbol{\kappa}_{Bi}, \boldsymbol{\kappa}_{Tri}) = T_{Bi}^{MT}(a_{Bi}, q, \dot{q}; \boldsymbol{\kappa}_{Bi}) - T_{Tri}^{MT}(a_{Tri}, q, \dot{q}; \boldsymbol{\kappa}_{Tri}) \quad (57)$$

$$T_{Bi}^{MT}(a_{Bi}, q, \dot{q}; \boldsymbol{\kappa}_{Bi}) = \frac{F_{Bi}^{MT}(a_{Bi}, \tilde{l}_{Bi}^{M}, \tilde{v}_{Bi}^{M}, \tilde{l}_{Bi}^{T}; \boldsymbol{\kappa}_{Bi})l_{2,Bi} \sin(q) \, l_{1,Bi}}{l_{Bi}^{MT}(q)} \quad (58)$$

$$T_{Tri}^{MT}(a_{Tri}, q, \dot{q}; \boldsymbol{\kappa}_{Tri}) = \frac{F_{Tri}^{MT}(a_{Tri}, \tilde{l}_{Tri}^{M}, \tilde{v}_{Tri}^{M}, \tilde{l}_{Tri}^{T}; \boldsymbol{\kappa}_{Tri})l_{2,Tri} \sin(q) \, l_{1,Tri}}{l_{Tri}^{MT}(q)}. \quad (59)$$

**Appendix E: Study on Out-of-Distribution predictivity of the MR PI-RNN training**

To investigate the performance of the MR training strategy on predictions made on trials that are out-of-distribution of the training data, two combinations of training and testing trials from the data used in the verification example (Section 4) were studied.

Table 6: Training and Testing data description for the Out-of-Distribution predictivity study.

	Training Data Trials	**Testing Data Trial**
Set A	2, 3, 4, 5	1
Set B	1, 2, 3, 4	5

The setting of this study is the same as described in Section 4. The MSE and R^2 scores after applying 1-, 2- and 3-scale MR PI-RNN training are shown in Figs. 12-13. In both tests, the prediction accuracy of the testing trial increases with the number of training scales. Such trend is more apparent when the sEMG signals contain more high-frequency content (from Case 1 to 3). The results demonstrate the effectiveness and the ability for out-of-distribution predictions of the proposed approach.
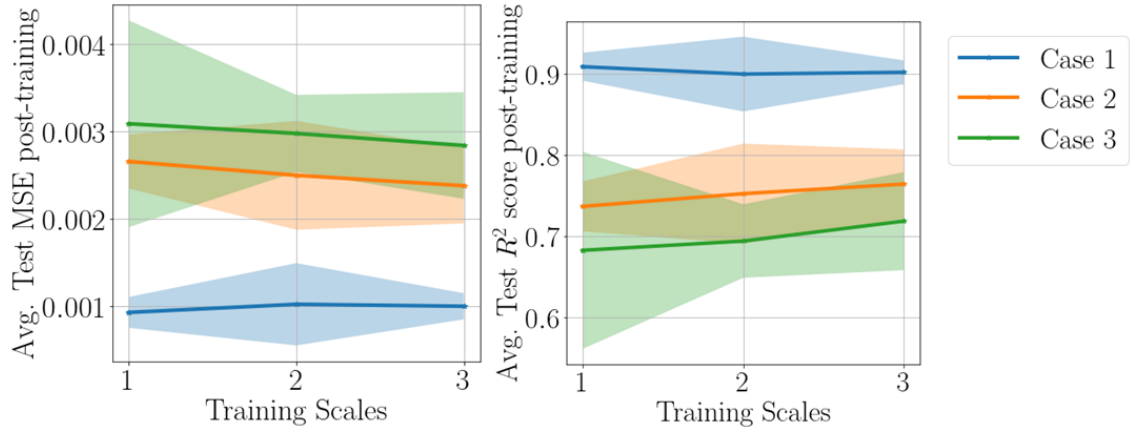
Fig. 12 The average test MSE and R^2 score post MR training for Set A are shown. The shaded area indicates one standard deviation from the mean (solid line), in both figures.
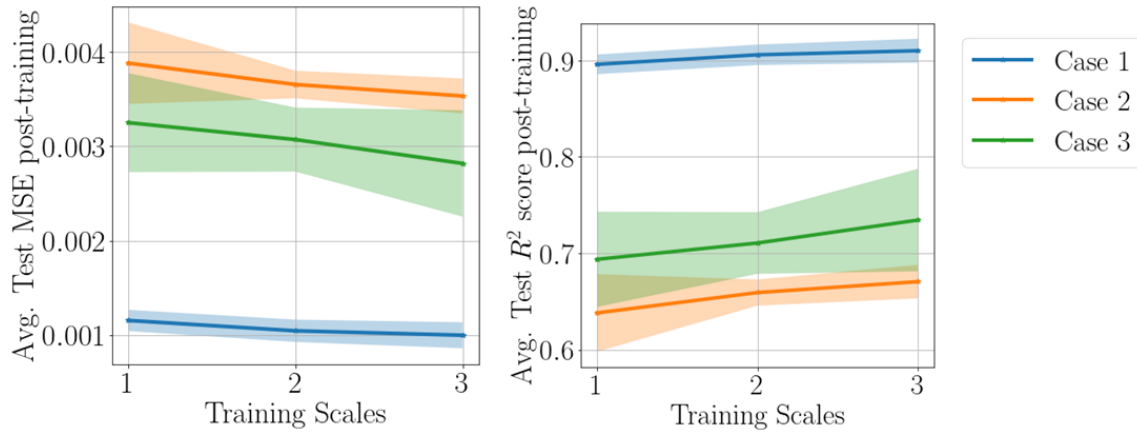
Fig. 13 The average test MSE and R^2 score post MR training for Set B are shown. The shaded area indicates one standard deviation from the mean (solid line), in both figures.