Medical & Biological Engineering & Computing
https://doi.org/10.1007/s11517-026-03519-x

ORIGINAL ARTICLE

PerSiVal: deep neural networks for pervasive simulation of an activation-driven continuum-mechanical upper limb model

David Rosin[1,5] · **Johannes Kässinger**[2,5] · **Xingyao Yu**[3,5] · **Michael Sedlmair**[3,5] · **Okan Avci**[4] · **Christian Bleiler**[1] · **Oliver Röhrle**[1,5]

Received: 17 December 2024 / Accepted: 10 January 2026

Abstract This paper introduces a novel densely connected neural network architecture designed for the pervasive visualisation of musculoskeletal system simulations. These simulations are built upon continuum-mechanical frameworks, which effectively integrate the diverse structural and physiological properties of the musculoskeletal system. A significant drawback of continuum-mechanical musculoskeletal models is their substantial computational resource requirement, making them difficult to transfer to/visualise the results on resource-poor systems like augmented reality or mobile devices. Such technologies, however, will be crucial for future advancements in human-machine interaction, surgical support tools, or physiotherapy. We use an activation-driven five-muscle continuum-mechanical upper limb model to obtain the activation-induced deformations of the respective muscles. Exemplified on the *m. biceps brachii*, we fit a sparse grid surrogate to capture the surface deformation and train a deep learning model that is subsequently used in our real-time visualisation. Based on the activation levels of the five muscles, the result of our trained neural network leads to an average positional error of 0.97±0.16 mm, or 0.57±0.10% for the 2809 mesh nodes of the *m. biceps brachii*'s surface. With the novel deep neural network model, we achieved evaluation times for the *m. biceps brachii*'s surface deformation of 9.88 ms on CPU-only architectures and 3.48 ms on architectures with GPU support. This leads to theoretical frame rates of 101 fps and 287 fps, respectively. The combination of surrogates and deep neural networks presented here succeeds as a proof-of-concept for real-time visualisation of a complex musculoskeletal system model, and does not rely on the inherent characteristics of the musculoskeletal system, and, hence, is also applicable to other real-time visualisations of complex meshed models in other applications.

Keywords Pervasive simulation · Upper limb biomechanics · Continuum mechanics · Surrogate modelling · Deep learning

1 Introduction

Simulations enable the analysis of complex systems, even if they are inaccessible from an experimental point of view, be it for technical, conceptual, or ethical reasons. One such example is identifying tissue deformation due to forces exerted by or imposed on our musculoskeletal apparatus. Such data is essential for tools intended to improve or contribute to computer-aided surgery, in particular in the field of neuromuscular interventions. They rely on accurate and instantaneous predictions of a tissue's response to external manipulation [5, 50]. Other examples are predicting joint movement due to muscle activation in the fields of rehabilitation, general physiotherapy, or training [22].

If such applications appeal to biomechanical models at all, they typically appeal to simplifying lumped-parameter

✉ David Rosin
rosin@imsb.uni-stuttgart.de

1 Institute for Modelling and Simulation of Biomechanical Systems, University of Stuttgart, Pfaffenwaldring 5a, 70569 Stuttgart, Germany

2 Institute for Parallel and Distributed Systems, University of Stuttgart, Universitätsstrasse 38, 70569 Stuttgart, Germany

3 Visualization Research Center VISUS, University of Stuttgart, Allmandring 19, 70569 Stuttgart, Germany

4 In-Silico Human Modeling, Fraunhofer IPA, Nobelstrasse 12, 70569 Stuttgart, Germany

5 Stuttgart Center for Simulation Science (SC SimTech), University of Stuttgart, Pfaffenwaldring 5a, 70569 Stuttgart, Germany

Published online: 03 February 2026

models [27, 63] such as those typically used in multi-body systems, e.g. [40]. Even though they can provide results in real-time [20, 59], they are not suitable for the above-mentioned applications. Continuum-mechanical models are particularly useful for the volumetric description of tissue deformation. Depending on their granularity, i.e., level of detail and complexity, e.g., inclusion of neurological aspects [cf., 2, 9], considerable computing times might be needed – even if significant computational resources, such as supercomputers, are available [11, 37, 38]. Continuum-mechanical-based Finite Element (FE) simulations have an advantage, in particular over state-of-the-art Hill-type models, that they can take the spatial heterogeneities and multi-physical phenomena into account. Within such a framework, the governing equations of finite elasticity, realistic muscle geometries, and consistent initial and boundary conditions are discretised using the FE method [17, 47]. The solution of the resulting nonlinear system consists of the displacements of each nodal point of the FE mesh.

Existing real-time and fully three-dimensional volumetric biomechanical simulations are proposed, for example, by Bro-Nielsen and Cotin [12] for surgical interventions involving the uterus, or by Mendizabal et al. [39], a recent work on liver deformation. Besides individualised approaches to simulate biomechanically realistic and three-dimensional tissue deformations in real-time, different modelling environments have been developed over the last years, for example, Artisynth, which implements a hybrid approach between volumetric and lumped-parameter model parts to focus computational resources [36], or SOFA, which facilitates modularity with regards to more or less complex solvers and linear constitutive models [5]. Both aim to optimise performance but don't adhere to any specific real-time criterion like a minimum frame rate, and compromise of model fidelity for the sake of performance. Predicting and visualising muscle deformations in real-time based on continuum-mechanical musculoskeletal system modelling that also takes into account the muscular activation level, has not been achieved up till now. Neither exist Virtual/Augmented Reality (VR/AR) environments that integrate such realistic, high-fidelity, biomechanical musculoskeletal system models into interactive systems. Most VR/AR approaches build on a gaming engine, or appeal to techniques typically developed for animations neglecting the underlying biophysical principles [6].

With this research, we aim to overcome these limitations. We hypothesise that on-body VR/AR visualisation of realistic muscle deformations resulting from a three-dimensional, volumetric biomechanical musculoskeletal system model is feasible in real-time, i.e., by generating more than 30 frames per second – a commonly used lower bound for interactive systems to provide a smooth visual experience [cf. 19, 28, 14]. Making simulation technology accessible "anytime, anywhere" often on resource-constrained devices like augmented reality or mobile systems, is the subject of the research field of pervasive computing. In this context, pervasive computing means distributing computations and adopting model accuracy to achieve the required performance regardless of the computational hardware, e.g., on a mobile device or a server infrastructure. In this sense, we focus on visualising realistic muscle deformation as it happens, where it happens, according to the camera-tracked movements of the person whose body the simulation results are overlayed upon on a given viewing device.

Aside from proof of concept, our target application serves more application-specific scenarios designed for medical staff or therapeutic applications. Reliance on expensive, specialised hardware stands in direct opposition to this. Hereby, surrogate modelling aims to offer an accurate prediction of the input-output behaviour of a quantity of interest in a model, at a lower computational cost than evaluating the model itself. Beyond more traditional surrogate modelling methods, such as model order reduction techniques, Deep Learning (DL) approaches have been established in an effort to find abstract, real-time capable surrogate models for biomechanical simulations, as well as FE simulation in general [18, 29, 46, 54]. Deep Learning methods require large amounts of training data, which can, for example, be generated using offline simulations. Here, offline refers to running the necessary simulations completely independent of the target application and on as large high-performance computing systems as needed.

One computational method to efficiently approximate complex and expensive simulations or systems, used to accelerate tasks like optimisation, uncertainty quantification, or sensitivity analysis, is to use surrogate models [4, 13, 21, 53]. Within this work, we combine several surrogate modelling techniques. Specifically, we adapt the work of Valentin et al. [58] to muscle surface deformation and use the resulting surrogate's output, which is based on a high-fidelity biomechanical FE musculoskeletal system model of the upper limb, as training data for a DL model (see also [33]). In other words, we simulate a high-fidelity model and use its output data to construct surrogate and DL models to achieve the maximum accuracy and evaluation speed possible with the respective and available hardware infrastructure. Further, we demonstrate that the performance is sufficient that the utilised high-fidelity model of the human upper limb can be visualised in real-time and on-body, using resource-constrained mobile devices such as VR/AR hardware. In this context, we assume that data about a person's arm pose is provided in real-time and the neurological input to each muscle (one homogenised activation level per muscle) is known, e.g., as a solution of an inverse problem that

appeals to a sparse grid (SG) as a surrogate model (sparse grid surrogate for short).

2 Methods

Before going into the surrogate modelling process and its individual components, first, the continuum-mechanical model of the upper arm will be described in some detail, as it is both the source of the data required for the proposed surrogate modelling/deep learning methods, as well as the starting point for all further considerations. The model was provided by Fraunhofer IPA and is presented in more detail in [7].

2.1 Continuum-mechanical model

The continuum-mechanical model of the upper limb consists of two main parts: a geometrical model discretised using the FE method and an appropriate constitutive formulation for the stress tensor describing the mechanical behaviour of the skeletal muscle tissue. In brief, in continuum mechanics a body $\mathcal{B}$ is considered as a coherent manifold of material points (particles) $\mathcal{P} \in \mathcal{B}$. For later reference, we denote the body's surface as $\partial\mathcal{B} \subset \mathcal{B}$. The material points $\mathcal{P}$ are parametrised by means of a position vector X in the undeformed reference configuration and the placement function $\boldsymbol{\chi}(\mathcal{P}, t) = \boldsymbol{\chi}(\mathrm{X}, t) = \mathrm{x}(\mathrm{X}, t)$ establishes the mapping (point map) between the reference configuration and its current configuration with coordinates x. Consequently, the displacement vector at time t is defined as $\mathrm{u}(\mathrm{X}, t) = \mathrm{x}(\mathrm{X}, t) - \mathrm{X}$. Further, the deformation gradient (line map) F is defined as

$$\mathrm{F}(\mathrm{X}, t) = \frac{\partial \boldsymbol{\chi}(\mathrm{X}, t)}{\partial \mathrm{X}} = \frac{\partial \mathrm{x}(\mathrm{X}, t)}{\partial \mathrm{X}} \quad . \tag{1}$$

For later use, we further introduce the right Cauchy-Green deformation tensor $\mathrm{C} = \mathrm{F}^T\mathrm{F}$.

The geometrical (FE) model of the upper limb contains three bones and five muscles (see Fig. 1). The bones are modelled as rigid bodies and are namely the *humerus*, *radius*, and *ulna*. *Radius* and *ulna* revolve around a hinge joint, connecting them to the lower head of the *humerus*. The muscles actuating the joint are the *m. biceps brachii* (biceps), *m. brachialis* (brachialis), and *m. brachioradialis* (brachioradialis) for flexion as well as the *m. triceps brachii* (triceps) and *m. anconeus* (anconeus) for extension. The distal end of the humerus is fixed and aligned with the z-axis. The anatomical model of our upper limb (Fig. 1) is based on the Visible Human dataset [2] and discretised using the FE method. The meshing was done with the pre-postprocessor

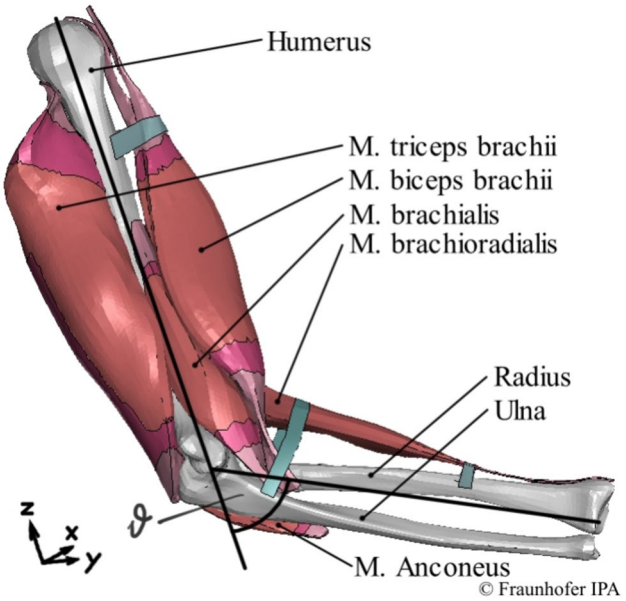

Fig. 1 Finite-element model. Rendering of the finite-element upper-limb model, with the definition of the joint angle ϑ. The colours help differentiate between different tissue compositions. For each individual muscle, moving from the muscle belly towards the tendon, these are: pure muscle (brown-red), a muscle-tendon-mix (magenta), pure tendon (light red) and bands mimicing connective tissue (light blue)

ANSA. Special attention was given to ensuring sufficient accuracy of element density and element quality. The entire muscle-tendon-system consists of 638 737 tetrahedral, linear Lagrange FE elements with 4 nodes each. The biceps by itself, which serves as our proof-of-concept example in this paper, is discretised with 68 292 elements. For the mutual interaction between muscles and muscle with bone, a mortar surface-to-surface contact formulation is applied. The proximal tendons are fixed to the ulna and radius with tied node-to-surface contact formulation.

The constitutive material description used within this work for the second Piola-Kirchhoff stress tensor S is based on a formulation using the right Cauchy Green tensor that is equivalent to the transversely isotropic muscle-tendon-description proposed by Röhrle et al. [49]. To describe the mechanical behaviour of the skeletal muscles constitutive behaviour, they use a phenomenological approach that is based on the principal invariants $I_1 = \mathrm{tr}[\mathrm{C}]$, $I_2 = \mathrm{tr}[cof[\mathrm{C}]]$, $I_3 = \det[\mathrm{C}]$ and the current (muscle) fibre stretch $\lambda_f = \sqrt{\mathrm{a} \cdot \mathrm{a}} = \sqrt{\mathrm{tr}[\mathrm{MC}]}$ with the structure tensor $\mathrm{M} = \mathrm{a}_0 \otimes \mathrm{a}_0$, the referential fibre direction unit vector a_0 and "$\otimes$" denoting the dyadic product. The vector a_0 relates to its current counterpart via $\mathrm{a} = \mathrm{F}\,\mathrm{a}_0$. The stress S is additively split into an isotropic and an anisotropic term, i.e., $\mathrm{S} = \mathrm{S}_{\mathrm{iso}} + \mathrm{S}_{\mathrm{aniso}}$, and the anisotropic part is further split to take into account passive anisotropic tissue, e.g., tendon, and active contributions of skeletal muscle tissue, [cf. 32]:

$$\mathrm{S} = \mathrm{S}_{\mathrm{iso}} + (\mathrm{S}_{\mathrm{passive}} + \alpha\gamma_M \mathrm{S}_{\mathrm{active}})(1-\gamma_{ST}), \tag{2}$$

where $\alpha \in [0,1]$ is the muscle activation, as a percentage of the maximum possible active stress, $\gamma_M, \gamma_{ST} \in [0,1]$ are factors to adjust the tissue compositions, e.g. muscle with $\gamma_M, \gamma_{ST} > 0$), tendon ($\gamma_M = 0, 0 < \gamma_{ST} \leq 1$), and fat ($\gamma_M = \gamma_{ST} = 0$). The isotropic part constitutes a nearly incompressible Mooney-Rivlin-type material [cf. 42, 48]:

$$\mathrm{S}_{\mathrm{iso}} = (B_1 \mathrm{I} + B_2 \mathrm{C} + B_3 \mathrm{C}^{-1}) + k\,(I_3^{1/2} - 1)\, I_3^{1/2}\, \mathrm{C}^{-1}, \tag{3}$$

where k is the so-called bulk modulus [16],

$$B_1 = 2c_1 I_3^{-1/3} + 2c_2 I_3^{-2/3} I_1, \tag{4}$$

$$B_2 = -2c_2 I_3^{-2/3}, \text{ and} \tag{5}$$

$$B_3 = -2/3 c_1 I_3^{-1/3} I_1 - 4/3 c_2 I_3^{-2/3} I_2. \tag{6}$$

with material parameters c_1 and c_2. As described in [49], the passive term $\mathrm{S}_{\mathrm{passive}}$ of the anisotropic stress is chosen to be:

$$\mathrm{S}_{\mathrm{passive}} = \begin{cases} \frac{c_3}{\lambda_f^2}\left(\lambda_f^{c_4} - 1\right)\mathrm{M} & \text{for } \lambda_f \geq 1\,, \\ 0 & \text{else}\,, \end{cases} \tag{7}$$

where c_3 and c_4 are material parameters associated with the anisotropy of the material. The active part $\mathrm{S}_{\mathrm{aniso}}$, is given by:

$$\mathrm{S}_{\mathrm{active}} = \begin{cases} \frac{\sigma_{\max}}{\lambda_f^2} \exp\left(-\left|\frac{\lambda_f/\lambda_f^{\mathrm{opt}} - 1}{\Delta W_{\mathrm{asc}}}\right|^{\nu_{\mathrm{asc}}}\right)\mathrm{M} & \text{for } \lambda_f \leq \lambda_f^{\mathrm{opt}} \\ \frac{\sigma_{\max}}{\lambda_f^2} \exp\left(-\left|\frac{\lambda_f/\lambda_f^{\mathrm{opt}} - 1}{\Delta W_{\mathrm{desc}}}\right|^{\nu_{\mathrm{desc}}}\right)\mathrm{M} & \text{for } \lambda_f > \lambda_f^{\mathrm{opt}}\,, \end{cases} \tag{8}$$

where $\sigma_{\max}$ is the maximum active stress a muscle can generate at its optimal fibre length λ_f^{opt}. Furthermore, W_{asc} and ν_{asc} define the ascending part of the force-length-relation of the muscle, while W_{desc} and ν_{desc} are the corresponding counterparts for the descending part. The material parameters are summarised in Table 5 in the Appendix.

The constitutive material model defined in Eqs. 2-8 has been implemented by the Fraunhofer IPA as a user-material in the commercial FE software LS-DYNA[1]. For further details on the originally incompressible formulation of the constitutive model see [49, 52], and [7] for additional information on this implementation for the upper limb model, as well as the FE model itself. An implicit solver is used to solve the appropriate weak form of the linear momentum balance (the governing equation)

$$\mathrm{div}[\boldsymbol{\sigma}] + \rho\,\mathrm{g} = \rho\,\ddot{\mathrm{x}}, \tag{9}$$

in spatial configuration. The stress tensor S from Eq. 2 is connected to the Cauchy stress tensor in Eq. 9 via $\boldsymbol{\sigma} = \det[\mathrm{F}]^{-1}\,\mathrm{F}\,\mathrm{S}\,\mathrm{F}^T$. Further, ρ is the material density in the current configuration and g denotes acceleration due to gravity. We expect sufficiently slow movements, $\ddot{\mathrm{x}} \approx 0$, so that the problem becomes quasi-static. To ensure nearly incompressible deformations, k is chosen sufficiently large.

The continuum-mechanical model described in this section constitutes a function $\boldsymbol{\alpha} \rightarrow \mathrm{x}(\mathcal{P})$, $\forall\, \mathcal{P} \in \mathcal{B}$ and $\forall\, \alpha_i \in [0,1]$, $i = 1,..,5$, where α_i refers to the activation levels of the individual muscles in the model, while $\boldsymbol{\alpha}$ is a vector containing all five activation values. For the subsequent surrogate modelling we will - without loss of generality - focus on the biceps surface specifically.

2.2 Sparse grid surrogate

With respect to the elbow joint angle, the upper limb model represents an overdetermined system. Consequently, an optimisation is needed to solve the inverse problem of finding the muscle activation levels necessary to reach a given pose. The original FE model cannot be used to handle this process in a reasonable time span, not even on high-performance hardware. Tests on a cluster with 64 CPUs, with 3 GHz each, and 1 TB of RAM showed evaluation times of up to 4 minutes per deformation state. Instead, FE model evaluations were restricted to 1053 states, uniformly distributed across the 5D parameter space of all activation levels. The number of states and their distribution were chosen to fit the support for a uniform SG. We use this surrogate to interpolate the nodal displacements corresponding to these known 1053 activation states from the FE model in order to evaluate the displacement for any activation $\alpha_i \in [0,1]$. The SG only interpolates in the activation domain, not in space. Strictly speaking, we use an ensemble of SGs, one per node in the FE mesh. For the purposes of this paper, we simply refer to this ensemble as a whole as the SG surrogate, which constitutes a function $\alpha \rightarrow \mathrm{x}(P_{\mathrm{mesh}})$, $\forall\, P_{\mathrm{mesh}} \in \partial\mathcal{B}_{\mathrm{biceps}}$, where P_{mesh} refers to mesh node, rather than material points $\mathcal{P}$. The size of the support of a SG is characterised by its level l. A higher level results in a more fine-grained grid, allowing for more accurate interpolation. Choosing a level is thus a trade-off between computational cost and accuracy. Based on the previous findings by Valentin et al. [58], a

[1] https://lsdyna.ansys.com

regular SG of level $l = 2$ was deemed dense enough to achieve sufficient approximation accuracy. Each activation level of one of the five muscles in the model constitutes one dimension of the SG. Spanning this parameter space with an SG of level 2 requires the aforementioned 1053 unique activation combinations. The interpolation between the support points is based on cardinal B-splines of degree $p = 3$. A B-spline $b^p : \mathbb{R} \to \mathbb{R}$ is defined recursively by:

$$b^0(x) = 1_{[0,1)}(x), \quad b^p(x) = \int_0^1 b^{p-1}(x-y)dy \quad . \tag{10}$$

The hierarchisation of these splines is given as the affine transformation

$$\varphi^p_{l,i}(x) = b^p\left(\frac{x}{h_l} + \frac{p+1}{2} - i\right), \quad h_l = 2^{-l} \quad . \tag{11}$$

For SGs of dimension *d*, multivariate, hierarchical B-splines are obtained by computing a tensor product

$$\varphi^{\mathrm{p}}_{\mathrm{l,i}}(\mathrm{x}) = \prod_{t=1}^{d} \varphi^{p_t}_{l_t,i_t}(x_t), \quad \mathrm{l} \in \mathbb{N}^d, \ \mathrm{i} \in I_{\mathrm{l}} = I_{l_1} \times, ..., \times I_{l_d} \quad , \tag{12}$$

with corresponding grid-points

$$\mathrm{x_{l,i}} = (x_{l_t,i_t})_{t=1,..,d}, \quad x_{l_t,i_t} = i_t \cdot h_{l_t} \quad . \tag{13}$$

In the multivariate case, and encompass all indices and levels associated with a given quantity, as both of those are defined per input dimension and can vary between them. Valentin and Pflüger [57] showed that for odd *p* the nodal subspace V_l^p of a regular SG of level *l* and dimension *d* can be described by the direct sum:

$$V^{\mathrm{p}}_{\mathrm{l}} = \bigoplus_{\|\mathrm{l}\|_1 \leq n+d-1} \mathrm{span}\{\varphi^{\mathrm{p}}_{\mathrm{l,i}} | \mathrm{i} \in I_{\mathrm{l}}\} \tag{14}$$

The contribution of each subspace to the approximation $\hat{f}$ of a given function $f : [0, 1] \to \mathbb{R}$ is controlled by weighting-coefficients, the so-called hierarchical surpluses $\alpha_{l',i'}$, and defined by

$$\hat{f}(\mathrm{x}) = \sum_{l'=0}^{l} \sum_{i' \in I_{l'}} \alpha_{l',i'} \varphi_{l',i'}(\mathrm{x}) \ , \tag{15}$$

with

$$\hat{f}(x_{l,i}) = f(x_{l,i}) \ \ \forall i \in \left[0, 2^l\right] \ . \tag{16}$$

2.3 Neural network surrogate

With sparse-grid-surrogate evaluation times of up to 100 ms for the visualisation of the biceps' surface, the SG surrogate is notably faster than the original FE simulation but not fast enough for real-time visualisation. Thus, results from SG interpolation are used as training data for a real-time-capable deep learning surrogate. To predict the activation- and stretch-induced deformation of the biceps' surface in real-time, a densely connected feed-forward neural network (NN) is employed. Note the NN surrogate is not supposed to be a one-to-one replacement for the SG. As the relation between joint angle and muscle activation input might not be bijective, i.e., multiple joint angles might relate to the same muscle activation levels, we additionally desire "energy efficiency of the musculoskeletal system," i.e., penalising sets of activation levels with high values.

Within this work, we achieve this by defining the following cost function for the optimisation:

$$\boldsymbol{\alpha}^{\mathrm{k}}_{\mathrm{opt}} = \underset{\boldsymbol{\alpha}^{\mathrm{k}} \in [0,1]^5}{\mathrm{argmin}} \left(\left| \Theta^{\mathrm{k}} - \vartheta^{\mathrm{k}}(\boldsymbol{\alpha}^{\mathrm{k}}) \right| + 0.05 \cdot \sum_{i=1}^{5} \alpha^{\mathrm{k}}_i \right) \quad , \tag{17}$$

where Θ^{k} is the target elbow angle and $\vartheta(\boldsymbol{\alpha}^{\mathrm{k}})$ is the in silico determined elbow angle due to activation parameters $\boldsymbol{\alpha}^{\mathrm{k}}$. The scaling of the sum of the activation levels was chosen through a series of optimisation test runs, to determine how much the sum of activations needs to be discounted to not favor low activations to the point of missing the given target joint angle Θ^{k}. With a factor of 0.05, the target joint angle was achieved with an accuracy of at least $0.001°$ in all test cases.

To speed up the activation-joint-angle-relationship calculations during the online phase, we evaluate Eq. 17 for a set of admissible activation parameters ($10° \leq \Theta^{\mathrm{k}} \leq 150°$) and subsequently use the $\boldsymbol{\alpha}^{\mathrm{k}}_{\mathrm{opt}}$ within a SG surrogate for the deformation of each mesh node on the muscles' surface. As we consider quasi-static conditions, we omit superscript *k* for the time instance from now on. The deformation is then used to train a NN surrogate, i.e., it is trained to approximate a function $\boldsymbol{\alpha}_{\mathrm{opt}} \to \mathrm{u}(P_{\mathrm{mesh}}), \forall \ P_{\mathrm{mesh}} \in \partial\mathcal{B}_{\mathrm{biceps}}$. This is obtained by evaluating the SG surrogate for all the mesh-nodes P_{mesh} at $\boldsymbol{\alpha}_{\mathrm{opt}}$, and subtracting the known reference coordinates of the given node $\mathrm{X}(P_{\mathrm{mesh}})$ from the result. All these evaluations and the training of the network can be done during the offline phase. The reference configuration is added to the bias of the last layer in the NN post-training, to convert from the displacement back to the Cartesian coordinates of the mesh-nodes. With this the NN surrogate provides the desired function $\boldsymbol{\alpha}_{\mathrm{opt}} \to \mathrm{x}(P_{\mathrm{mesh}}), \forall \ P_{\mathrm{mesh}} \in \partial\mathcal{B}_{\mathrm{biceps}}$.

The NN was implemented using the open-source API Keras for TensorFlow in Python [1, 15].

2.3.1 Neural network architecture

The NN architecture needs to bridge the gap between the number of inputs $n_{in} = 5$, i.e., the number of muscle activation parameters of the FE model, and the number of outputs $n_{out} = 2809$. In lack of any immediate benefit from more specialised layer-types as per the data, the NN consists entirely of densely connected layers. While there are three Cartesian coordinates to consider for each node, the coordinate directions are trained separately, each using an identical architecture, receiving the same input. How the training was conducted will be described in more detail in Section 2.3.2. The width w of each layer, i.e., the number of neurons per layer, increases exponentially from input to output of the NN. While funnel-like shapes are fairly common in deep learning, e.g. in autoencoders [33, 34], this architecture was not designed in reference to an existing one. See section 3.1 for more details on the development process.

The parameters of the exponential function are the number of hidden layers n_{hidden}, n_{inputs}, n_{outputs}, and n_{mid}, which specifies the width to be reached half-way between the input- and output-layer:

$$w(n) = a \cdot e^{b \cdot n} + c \ , \ \text{with} \tag{18}$$

$$a = \frac{n_{\text{mid}}^2 - n_{\text{inputs}}^2}{n_{\text{outputs}} - 2 \cdot n_{\text{mid}} - n_{\text{inputs}}} \ , \tag{19}$$

$$b = 2 \cdot n_{hidden} \cdot \log\left(\frac{n_{\text{outputs}} + n_{\text{inputs}}}{n_{\text{mid}} + n_{\text{inputs}}}\right) \ , \tag{20}$$

$$c = n_{\text{inputs}} - a \ . \tag{21}$$

Figure 2 shows a visual representation of this NN architecture. The best training results were reached with the number of hidden layers $n_{hidden} = 5$ and the width of the mid-layer of the network $n_{\text{mid}} = 700$. Note, that there is not necessarily a layer with exactly the width n_{mid}, as is the case here. The exponential function spans from the input layer to the last hidden layer, putting the mid point between the second and third hidden layer (cf. Figure 2). The function parameters were chosen as small as possible, while maintaining sufficient accuracy, i.e., sufficiently low mean squared error (MSE), of the predicted coordinate output, by comparing NNs with varying values for these parameters.

2.3.2 Dimension-decoupled training

Three individual networks were trained, one for each coordinate dimension. Post-training, these individual models were combined into a single NN, with three channels, via network-graph redefinition, using the Keras functional API Fig. 2). Each of the three networks was trained using five-fold cross-validation [44], i.e., the training data was shuffled and divided into five partitions, one of which was used for validation and four for training for a certain number of epochs defined. Using five folds is a common choice, meant to ensure each individual fold still contains enough samples to be representative of the entire data-set. After each "fold", which partition is used for validation, and thus not shown to the model during training, changes. The networks were trained for 400 epochs on each fold, resulting in an overall training duration of 2000 epochs on the MSE, with the mean absolute error as an additional metric. Here, "mean" refers to a mean value across all surface nodes in a given

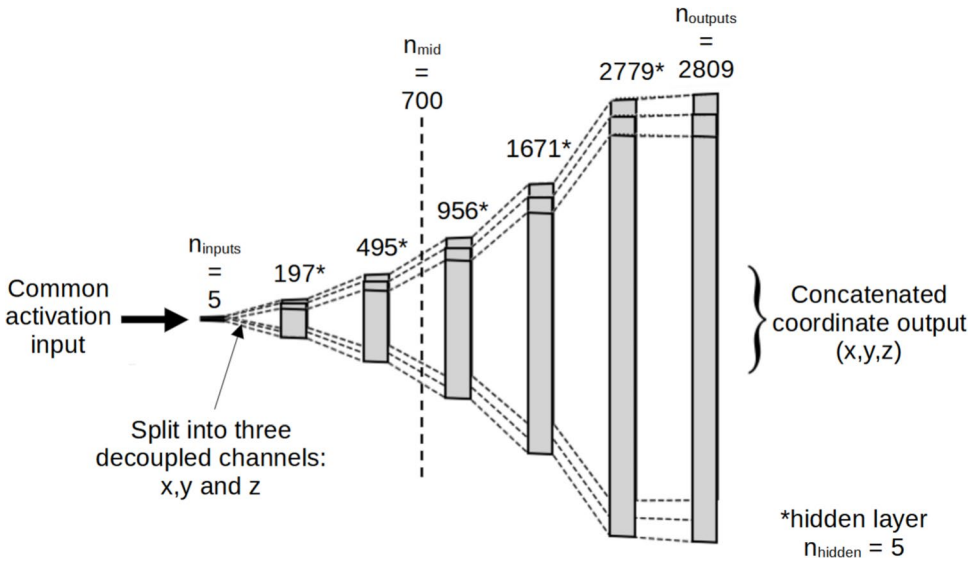

Fig. 2 Combined model. Combination of the three NNs for the x-, y- and z-coordinate, by defining a common input for all three networks (or three channels, once it is one model) and a concatenated output. The width of each layer is noted above it. The exponential function prescribing the layer width spans from the input to the last hidden layer, meaning n_{mid} is reached between the second and third hidden layer

Table 1 Comparison of three different prototype architectures for learning the X-coordinates of the biceps' mesh, based on in-training MAE of predicted node positions after ten epochs, and the number of trainable parameters

	T-shaped	Rectangular	Triangular
# of Parameters	17.154	71.056.494	26.103.046
MAE in mm	430	2.4	2.3

configuration. In the results and the discussion the mean absolute error (MAE), will be of particular interest, since it is a metric in millimeters, allowing for better physiological consideration of the errors magnitude. No custom schedule for the learning-rate, i.e. scaling of the size of each optimisation step applied to the network weights, was used. Additionally, Keras' "callbacks" functionality was used to select the version of the NN that performed best, i.e., has the lowest loss across all epochs. This was done to account for fluctuations in the in-training performance from epoch to epoch, meaning the last iteration of the NN was not necessarily the best-performing one. Keras' functional API allows for the redefinition of network graphs layer-by-layer. This was used to define a common input and output for the three networks, without needing to manipulate their tensors manually. The only manual weight-editing is the aforementioned addition of the reference configuration to the last layer's bias, converting the NN's output from displacement to Cartesian coordinates.

3 Results

3.1 Performance of the NN architecture

As the training data does not indicate an immediate benefit to using a more specialised layer type NN model, we chose fully connected layers for our NN. To find a suitable NN structure, we tested the following three rudimentary options to connect the five input parameters to the 2809 outputs:

1. changing the layer width as late as possible, resulting in a T-shaped NN
2. changing the layer width as early as possible, resulting in a rectangular NN
3. changing the layer width linearly as a function of the distance to the input, resulting in a triangular NN

With these networks as a starting point, we aimed to arrive at an architecture that best balances number of trainable parameters and prediction accuracy, without simply relying on trial and error. To be able to compare these three architectures, we generated prototypes with an arbitrary depth of 10 hidden layers.

For the initial prototyping stage, we assumed 10 epochs to be sufficient to check if a particular network shape would lend itself to the data. The results of a particular test is provided in Table 1. The T-shaped architecture performs notably worse than its competitors. The rectangular and triangular architectures are on par. However, the triangular architecture reaches this level of accuracy with less than half the number of trainable parameters of the rectangular one. Based on these findings, the exponential curve introduced in Eqs. 18 - 21 was set up to slim the layer width in the mid-section of the NN, while maintaining the basic shape of the triangular architecture. This slimmed network performed without loss in accuracy, with an MAE of 2.2 mm. Further, we reduced the number of hidden layers step-by-step. This was possible without loss of accuracy down to five hidden layers, before further reduction caused a notable decline in training accuracy, with an MAE of upwards of 50 mm.

These initial steps were taken while training on the nodal coordinates, not the deformation. Visualising results from direct coordinate prediction showed the biceps as a whole to noticeably drift sideways during contraction. Training on the deformation instead, and adding the known reference configuration of the model output to attain the coordinates of the deformed state, removed any visual drift from the predicted nodal positions. Further improvements were achieved with longer training times (cf. Figure 3). Furthermore, the dimension-decoupled training proofed to allow for a noticeably smaller NN model overall. Simply put, to reach the same level of accuracy, when training on the vectorised data directly, required more than three times the number of trainable parameters contained in a NN trained on only one of the three components of the coordinate vectors. In

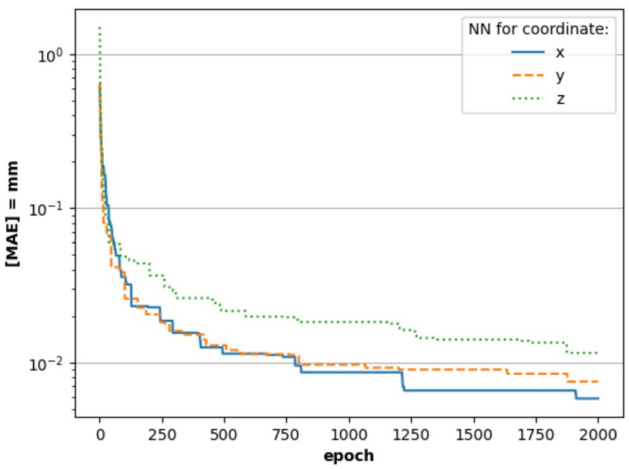

Fig. 3 Training history. In-training performance of the three coordinate-networks shown as the MAE of predicted nodal positions in mm over the number of epochs trained

fact, training the NN with the same number of hidden layers using the 3D data instead of considering each dimension by itself, the MAE did not decrease below 20 mm. Hence, three individual networks were trained, one for each coordinate dimension, and combined into a single NN via network-graph redefinition post-training.

3.2 Accuracy analysis

Figure 3 shows the performance of the current lowest loss model over the course of 2000 epochs of training. For reference, without the use of callbacks the model loss fluctuated between $1 \cdot 10^{-1}$ and $1 \cdot 10^{-2}$ mm of MAE. The initial drop in loss over the first 50 epochs experienced no fluctuations. After 2000 epochs, in-training accuracy showed a maximum absolute error (MaxAE) of 0.37 mm, which corresponds to 0.38% of the respective deformation at the respective location. The mean error is 0.014 mm.

Out-of-training performance usually refers to a NN generalisation capabilities. Since the input parameter space for this model is strictly bounded, this means evaluating activation combinations not contained in the training data set. We tested out-of-training performance by iteratively generating random activation vectors. New random activation vectors were continuously added and the NN's error was determined by comparing its output with the solution of the SG, which was assumed for this purpose to be the ground truth. This process terminated once the highest recorded MaxAE of the NN did not further increase for 300 consecutive iterations. We assume this criteria to be equivalent to having found the worst-case error. The results were a maximum absolute error (MaxAE) of 11.96 mm and a maximum relative error (MaxRE) of 13.28% (cf. first row of Table 2). Here, the MaxRE is not the converted MaxAE, but the overall recorded maximum relative error value. The highest mean absolute error (MAE) across all configurations was 2.15 mm, and the highest mean relative error (MRE) was 1.59%.

Random sampling will include activation vectors, which will not be encountered in the final application, as they are not an optimal choice for any given joint angle. In addition, we iteratively tested for random joint angles, for which the optimal activation vectors are computed, since activations not optimal to any joint angle should not be encountered in the end application. This test yielded a MaxAE of 3.75 mm and a MaxRE of 5.69%, a MAE of 0.97 mm and a MRE of 0.57%. See Table 2 for a summary of these results.

Table 2 Out-of-training accuracy. Summary of the out-of-training accuracy of the NN. Predicted nodal positions were analysed for random activations and optimal activations for random joint angles as NN inputs. Relative maximum errors were determined independently from the absolute values

	MaxAE (mm)	MaxRE (%)	MAE (mm)	MRE (%)
Random Activation	11.96	13.28	2.15	1.59
Random Angle	3.75	5.69	0.97	0.57

3.3 Error propagation

Since the NN surrogates are trained on data from the SG surrogate, not the FE simulation, i.e., training on data originating from an approximation, it is essential to study error propagation to ensure that the NN does not amplify the approximation error of the SG. To investigate this, the FE model was run for 25 activation combinations drawn randomly from a uniform distribution. Both the SG and the NN surrogate were evaluated for the same activation vectors, and the results of the subsequent comparison of both to the FE simulation are summarised in Table 3. The error of the NN with respect to the original FE simulation closely matches that of the SG.

3.4 Evaluation times and theoretical frame rates

In addition to analysing errors, the evaluation speed of the model was also tested. This is crucial for applications running on resource-limited hardware such as AV/VR goggles. Runtime was evaluated by once evaluating the runtime individually for x-, y- and z-coordinate and summing up the individual times, the runtime for the combined model (cf. Figure 2), and the runtime using a TensorFlow-Lite-version of the combined model. All runtime studies were performed on a CPU only infrastructure (Intel i5 with six cores and 3 GHz clock speed) and on a infrastructure with a GPU (Nvidia Geforce GTX 1650 with 4GB VRAM). The runtime results, averaged over 1000 evaluations, are given in Table 4.

On the CPU-only architecture, a theoretical frame rate of about 75 frames per second (fps) was achieved for the combined as well as the individual model approach. For execution on the GPU-supported architecture, the runtime differs, i.e., with the combined model being 0.57 ms faster than the sequentially executed models. Hence, the theoretically achievable frame rates are 246.91 fps and 287.36 fps, respectively. The results for the TensorFlow Lite model are

Table 3 Error propagation test. Summary of the accuracy of the SG and the NN tested on 25 random activations evaluated using the FE model. The NN was tested on the 1053 support points that were used to construct the SG from the FE-model as well

	MaxAE (mm)	MaxRE (%)	MAE (mm)	MRE (%)
SG	17.79	10.14	2.86	1.52
NN	18.28	9.38	2.84	1.45

Table 4 Comparison of evaluation times of different versions of the NN model, averaged over 1000 evaluations. The test was conducted with and without GPU-support, except for the TensorFlow Lite version, which is a CPU-only network format

Model version	Average runtime per execution	
	CPU-only architecture	GPU-supported architecture
three separate models	13.65 ms	4.05 ms
combined model	13.34 ms	3.48 ms
combined model Lite	9.88 ms	–

identical for GPU and CPU, with an evaluation time of 9.88 ms, i.e., a frame rate of 101.21 fps.

4 Discussion

Muscles undergo large deformations [45]. The average nodal deformation in our dataset of the biceps surface is 10.31 ± 8.62 mm, with a maximum deformation of 45.49 mm. Relative to that the accuracy of the NN prediction reported in Table 2 of 0.57 % is good (for further discussion see Section 4.1).

However a limitation we are facing with this proof-of-concept for any kind of clinical application is patient-specificness, not only with regards to model geometry, but also material parameters. Interpersonal variability due to age, gender or level of fitness can be considerable, no matter which part of the human body is being modeled [cf. 3, 23, 43]. This poses a challenge for the use in, e.g., surgery support tools. There it could be particularly useful to have access to quick evaluation of "what if...?"-scenarios. This would entail extending our surrogate modelling workflow to also consider parameters of our constitutive model as input, potentially enabling real-time assessment of the influence of parameter variation on the deformation output. This is currently beyond the scope of this contribution. That said, the results presented here still hold potential for applications outside of diagnostics, such as motion guidance or education in physiotherapy and training (also see Section 4.2).

In the intermediate results from the first 100 epochs of training, the accuracy of the predicted node coordinates differed substantially across the biceps' surface. More specifically, while visualising a cycle of arm flexion and extension, consistent areas of high absolute error. This did not correspond to a higher MRE than for other surface configurations. This mostly concerned configurations for which all flexors were at least partially activated. An example of a configuration exhibiting these hot-spots can be seen in Fig. 4, where the higher the MAE, the lighter the colour of the area. These regions, ordered by decreasing MRE, are namely the lower tendon, the medial side of the biceps' belly, as well as the more lateral, dorsal area of the muscle belly. The error distribution across the muscle equalised and diminished with prolonged training and eliminated all hot-spots after training was completed. Error hot-spots could be due to insufficient training or the anatomical complexity. Error hot-spots due to insufficient training can probably be neglected. This assumption is supported by the fact that the hot-spots level off with increased training times (cf. Figure 4). Inspecting the regions that exhibit high errors, one observes that they coincide with contact areas, e.g. the contact area between the biceps and other flexors, like the *M. brachioradialis* (BR). In the current model, the contact areas are relative small compared to the overall surface area of the muscle. Thus, the NN most likely does not focus on accurately learning the characteristics of the placement functions for the nodes within the contact areas compared to the rest of the muscle. This phenomenon will likely diminish if all muscles and the fat and skin layer are considered. In this case, the contact areas of the muscles are dominant. Implementation of a fat skin layer is desirable, but not essential for proofing the feasibility of the proposed method. In addition, the assumption that the main contributor to the error hot-spots is the complex anatomical geometry, is supported by the observation that the tip of the lower tendon of the biceps (attachment to

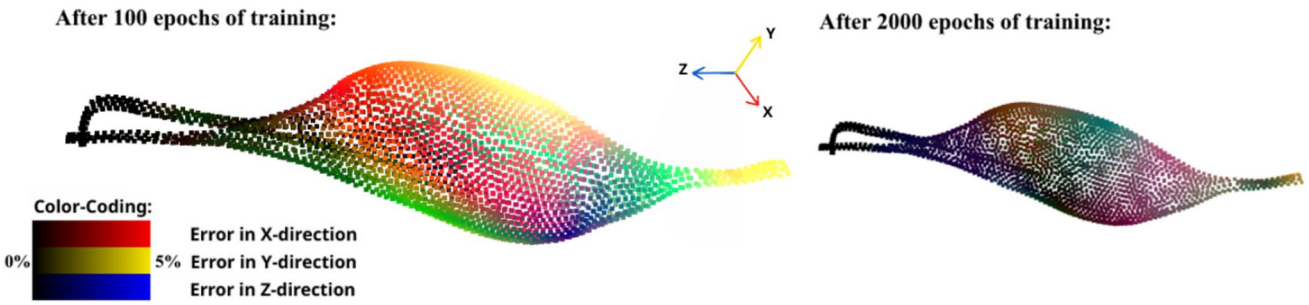

Fig. 4 Error "hot-spots". On the left: relative error analysis after 100 epochs of training, with colour-coding for x (red), y (yellow), z (blue). All three colour channels are normalised to 5% relative positional error. Bright yellow regions of comparatively high error appear in multiple configurations on the muscle belly and at the lower tendon (here on the right of the biceps). Post training, i.e. after 2000 epochs, the error is smaller overall, and more evenly distributed, as shown in small on the right

the bone) also exhibits comparatively higher errors. This is most-likely caused by both the complex geometry and the placement function for the respective nodes exhibit stronger nonlinearities.

We know from other deep learning models, especially in the case of classifiers, that rare cases pose a particular challenge for them. In such cases, data augmentation techniques are a common technique to make rare occurrences more prevalent in the data to increase their influence during training [35]. A similar idea could be applied here, e.g., by subdividing the NN model into parts with each specialising on one of these regions and nodes. However, since the hot-spots vanish given sufficient training time, we argue that data augmentation will not be necessary for our case. Future research with more complex musculoskeletal system models (in terms of number of muscle, multiple joints, and fat and skin layer) has to show how significant the muscle-to-muscle contact effect is pronounced. Ideally, this would also be accompanied by experimental validation, to supplement our in silico findings and further the clinical relevance of this research. However, as stated right at the beginning of this work, there is currently simply no experimental technique available to capture the deformation of entire muscles in vivo. Current developments in 3D muscle reconstruction using ultrasound may offer a solution to this, as well as open up possiblities for models with personalised geometries [51]. Introducing such patient specific elements into this pipeline, would also necessitate careful handling of both the data and the surrogate models, since they would have to be considered medically sensitive personal data.

4.1 Out-of-training performance

Out-of-training accuracy is within expectations. While the NN can not provide the level of accuracy of the original simulation or the in-training accuracy, an MaxRE of 5.69% and an MaxAE of 3.75 mm (cf. Table 2) are of no concern in the context of the visualisation of skeletal muscle deformation during motion, given they constitute the worst-case performance and are still small relative to the magnitude of the deformation. The MAE and MRE, which are one order of magnitude less than the worst-case performance, i.e., 0.97 mm and 0.57% respectively, support this statement. The level of accuracy measured out of training is good enough to suggest that the model has not gone into overfitting, where in-training performance is excellent, but out-of-training performance is considerably worse. For this particular iteration of this workflow, an easy case for reliability of the surrogate models can be made, in the sense that the entire input parameter space can be explored. For future versions, with more degrees of freedom, global exploration of the parameter space may become more difficult.

The additional results regarding error propagation shown in Table 3 show that inaccuracies of the SG dominate the overall error. These inaccuracies can not be disregarded as fringe effects, as they occur at mid-level values. Hence, an accurate SG model approximating the FE-model well is essential. To do so, we can either increase the number of SG support points overall for a finer uniform grid, or select them through an adaptive process. We are currently already working on a iterative, adaptive SG setup. In this implementation, the accuracy of the SG surrogate is tested against a validation data set, like the one used for the evaluation shown in Table 3, each iteration. The FE model is then prompted to run using activation levels the SG is estimated to be particularly inaccurate for, judging by its performance on the validation data. The results of this simulation can then be added to the SG support to increase its size until overall SG estimation error on the validation data set falls below a given threshold. For more details on this type of SG please refer for example to [56] and for the selection of new candidate support points to [31]. This would alleviate the the concerns brought up by the results shown in Table 3. As the NN surrogate can inevitably only be as accurate as the training data generated using the SG, this benefits the entire workflow.

Pervasive computing has the potential to further decrease the error and potentially the runtime. We envision to include additional, external computational resources on which one can run simulations of more complex and accurate models. The idea is to offload computations to other resources, dynamically at run-time, based on availability of resources and time constraints. The result is the best possible performance depending on the available hardware.

Overarching, deep learning models are black boxes with no formal way of predicting the accuracy or the error propagation if surrogate models are used. The result is highly dependent on the quality of the training data [41]. As we use here a continuum mechanical model, which is based on homogenising the micro-scale, and the FE method to discretize the underlying governing equations, our training data is subject to a discretisation error. This is, however, mitigated by the fact that FEM, continuum mechanics, as well as SG are formally well established [10, 61]. That said, in future iterations of this work the biomechanical modelling will be an area of focus, as additions like a fat skin layer surrounding the musculature and a more complex elbow joint, driven by the geometry of the joint (as compared to a hinge joint), could considerably improve the realism of the model and its dynamics. Note, that this would not necessitate changes to the surrogate modelling workflow presented here. It would improve the data piped through it.

The measured evaluation times (see Table 4) show that all tested versions of the model are real-time-capable. A

conversion to an optimised format, such as TensorFlow Lite, is preferable if no GPU is available. TensorFlow Lite is optimised for ARM CPUs, to better fit mobile devices, where they are commonly used. Thus, the results listed here likely do not show the peak performance of the TensorFlow Lite version of the model, since the Intel CPU, on which we determined the runtime, is a different architecture.

The visualisation of the overlay within the VR/AR environment and tracking any motion and applying it to the system will increase the overall compute time per frame. However, first tests of an implementation of the overlay combined with motion tracking were successful in providing a real-time experience. All versions of the model tested within this work, lie well below the guideline for real-time applications of 30 ms evaluation time, even on resource constraint devices [30], i.e., leaving us with computational resources to spare at the current accuracy and speed. In addition, see the work by Kässinger et al. [30] for a discussion on reducing computational cost, energy consumption and evaluation time on the end user's mobile device. As the work mentioned above shows a distributed variant of the NN architecture shown in Fig. 2, it has to be made clear that this a proof of concept. Other architectures exist that could learn the data used here. The current NN architecture was chosen as a good compromise between high accuracy for predicting the nodal deformation and low number of trainable parameters. More NN architecture exploration is needed in the future. A concern the aforementioned tests raised, is the computational effort needed for the motion tracking. In our current setup, the motion tracking algorithm clearly dominates the overall computational cost of the visualisation. Utilising light-weight and accurate motion tracking will be one of the major implementation challenges, as we include more and more muscles in the visualisation and computational cost increases for the run surrogate models.

4.2 Application scenario

These models will be implemented and extensively tested on resource-constraint devices, e.g. a distributed AR application on a Microsoft HoloLens 2 or tablets like the Apple iPad. To achieve this, firstly a second model trained on the relationship between the motion tracking data (elbow angle, angular velocity, and acceleration) and the optimal activation vectors will be introduced; secondly, once the output visualisation is overlayed onto a person, it will be supplemented with additional biomechanical information, e.g., graphs tracking the activation levels. This will be relevant for, e.g. coupling the model to a motion guidance system for physiotherapy or exercise.

For instance, in the physiotherapy setting, a doctor asks their patient to complete an arm movement following a guided path, shown to them in VR/AR. This model could help the patient to appreciate how their motions result from muscle contraction and aid the physiotherapist in explaining possible issues and solutions. Ideally, this could allow the patient to share in their therapist's intuition. The guidance could help provide feedback to the patient on what exactly the motion is they should try to achieve [25], and even continue to do so, when training unsupervised at home [55]. For a more in-depth discussion of such a system also see the following contribution by Yu et al. [62]. Considerations regarding AR applications in physiotherapy were made as part of a interdisciplinary tandem project, since implementing them requires a variety of backgrounds in order to succeed.[2] Perhaps even more importantly than guiding the patient, patient education has been shown to have a significant influence on therapy outcome. As a anatomically accurate, intuitive visual aid for the physiotherapist to inform the patient about what there musculature looks like, how it moves and works, this visualisation can directly improve the chances of therapeutical success. Doing these exercises at home, AR as a setup still allows for awareness of the physical surroundings alongside working with the simulation overlay. As we only consider slow movements, the risk posed by user error is low, but nevertheless, care should be taken, when instructing people on how to use these kinds of applications.

5 Conclusion and outlook

The aim of this work was to find a workflow that is suitable for visualising data from continuum-mechanical forward simulations in real-time, to further the development of pervasive simulation and open up continuum mechanics to a wider variety of applications. The presented proof-of-concept model is able to predict the surface configuration of the biceps, based on a vector of activation levels optimal for reaching a certain target elbow angle, within an average error of 0.97 mm at a frame rate of up to 287.35 fps on non-high-end hardware. This model thus fulfills our set criteria. Achieving this kind of accuracy in real-time with a musculoskeletal system model of this complexity is novel, and the opportunities for new kinds of visualisations based on complex, meshed FE simulation data, are numerous. As outlined in previous section, motion guidance for unsupervised physiotherapy and intuitive patient education are benefits this workflow can already provide. Additionally, as mentioned in Section 4, expanding this workflow to include key constitutive parameters and personalised geometries could greatly increase its applicability in a clinical setting. Also note that even taking the aim of pervasiveness aside,

[2] https://www.simtech.uni-stuttgart.de/exc/research/pn/pn7/pn7-1/

real-time prediction of the muscle's surface configuration on a non-high-end PC, is still an exciting result, considering the original FE simulation took minutes per configuration on 64 CPUs à 3GHz, with 1 TB of RAM. The use of a SG surrogate to generate the training data for the NN, has proved efficient for acquiring arbitrary amounts of training data within a feasible time frame. Accuracy of the SG still needs to be improved by optimising its support.

The NN architecture, as it stands now, might benefit from the use of convolutional layers, which we will investigate in the near future. In particular, compared to the densely connected architecture, convolutional layers would reduce the number of trainable parameters, by having some of the neural network's connections share weights, rather than having a unique weight for each connection. Whether or not this can be achieved while retaining the current accuracy, remains to be seen.

Further development of this model beyond the technical aspects will include incorporating it into new on-body interaction techniques, to not only further improve performance on a wider range of hardware, but also provide ease of use to the end-user. At the same time, NNs for the other muscles shown in Fig. 1 will be trained and integrated into the visualisation, to expand it to the model as a whole. Since only the output dimensionality changes between the muscles, as they consists of differently many nodes, this can be easily achieved with the workflow as is. In the same manner, the visualisation can be extended to include other types of nodal data, such as strains and stresses, or the interior of the muscles as well. Other types of scalar nodal data could be included without an increase in computational cost at run time, assuming only one type of data is being visualised at any given time. Whether or not real-time performance can be maintained when including the interior, given the considerable increase in the number of nodes and thus in the size of the NNs needed, remains to be seen.

Finally, we would like to note that the methods presented in this paper are neither limited to the muscle constitutive laws chosen in Section 2.1 nor to FE results in general. The presented sparse-grid-neural-network workflow is equally applicable to more complex, microstructural-based muscle constitutive laws [8] or activation mechanisms [32], as well as to other biological or man-made materials. Likewise, the method can be applied to any kind of mesh-based data, such as results coming from the finite-difference method.

Appendix A Constitutive parameters

Table 5 Model parameters. Constitutive parameters for the muscle-tendon-complex of the biceps

Parameter	Value	Source
c_1^M	$3.56 \cdot 10^{-2}$ MPa	[26]
c_2^M	$3.86 \cdot 10^{-3}$ MPa	
c_3^M	$3.57 \cdot 10^{-8}$ MPa	[64]
c_4^M	42.6	
c_1^T	2.31 MPa	[60]
c_2^T	$1.15 \cdot 10^{-6}$ MPa	
c_3^T	7.99 MPa	
c_4^T	16.6	
ΔW_{asc}	0.25	adapted from [24]
ΔW_{desc}	0.15	
ν_{asc}	3.00	
ν_{desc}	4.00	
λ^f_{opt}	1.35	
$\sigma_{\max}$	0.30 MPa	

Author Contributions **David Rosin:** Conceptualization, Methodology, Software, Validation, Data Curation, Writing - Review & Editing, visualisation **Johannes Kässinger:** Conceptualization, Writing - Review & **Editing Xingyao Yu:** Writing - Review & Editing **Michael Sedlmair:** Writing - Review & Editing, Supervision **Okan Avci:** Resources, Writing - Review & Editing **Christian Bleiler:** Writing - Review & Editing **Oliver Röhrle:** Writing - Review & Editing, Supervision, Funding acquisition

Funding Open Access funding enabled and organized by Projekt DEAL. This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075–390740016, the Stuttgart Center for Simulation Science (SimTech), the Bundesministerium für Bildung und Forschung (BMBF, German Federal Ministry of Education and Research) under Grant No. 01EC1907B and Grant No. 01EC1907D ("3DFoot"), and ERC-AdG 2021 #101055186 ("qMOTION").

Materials availability Not applicable

Declarations

Conflicts of Interest The authors have no conflicts of interest to declare.

Ethics statement Not applicable

References

1. Abadi M, Agarwal A, Barham P et al (2015) TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/
2. Ackerman MJ (1998) The visible human project. Proc of the IEEE pp 504–511. https://doi.org/10.1109/5.662875
3. Alfuraih AM, Tan AL, O'Connor P et al (2019) The effect of ageing on shear wave elastography muscle stiffness in adults. Aging Clin Exp Res 31(12):1755–1763. https://doi.org/10.1007/s40520-019-01139-0
4. Alizadeh R, Allen JK, Mistree F (2020) Managing computational complexity using surrogate models: a critical review. Res Eng Des 31(3):275–298. https://doi.org/10.1007/s40520-019-01139-0
5. Allard J, Cotin S, Faure F et al (2007) Sofa-an open source framework for medical simulation. In: Proc. medicine meets virtual reality (MMVR), IOP Press, pp 13–18, https://inria.hal.science/inria-00319416
6. Angles B, Rebain D, Macklin M et al (2019) Viper: Volume invariant position-based elastic rods. Proc Computer Graphics and Interactive Techniques pp 1–26. https://doi.org/10.1145/3340260
7. Avci O, Röhrle O (2024) Determining a musculoskeletal system's pre-stretched state using continuum–mechanical forward modelling and joint range optimization. Biomech Model Mechanobiol 23:1. https://doi.org/10.1007/s10237-024-01821-x
8. Bleiler C, Ponte Castañeda P, Röhrle O (2019) A microstructurally-based, multi-scale, continuum-mechanical model for the passive behaviour of skeletal muscle tissue. J Mech Behav Biomed Mater. https://doi.org/10.1016/j.jmbbm.2019.05.012
9. Böl M, Iyer R, Dittmann J et al (2019) Investigating the passive mechanical behaviour of skeletal muscle fibres: micromechanical experiments and bayesian hierarchical modelling. Acta Biomater. https://doi.org/10.1016/j.actbio.2019.05.015
10. Bonet J, Wood RD (1997) Nonlinear continuum mechanics for finite element analysis. Cambridge university press, http://thuvien.ued.udn.vn/handle/TVDHSPDN_123456789/41311
11. Bradley CP, Emamy N, Ertl T et al (2018) Enabling detailed, biophysics-based skeletal muscle models on hpc systems. Front Physiol 816. https://doi.org/10.3389/fphys.2018.00816
12. Bro-Nielsen M, Cotin S (1996) Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In: Proc. Computer graphics forum, Wiley Online Library, pp 57–66, https://doi.org/10.1111/1467-8659.1530057
13. Cheng K, Lu Z, Ling C et al (2020) Surrogate-assisted global sensitivity analysis: an overview. Struct Multidiscip Optim 61(3):1187–1213. https://doi.org/10.1007/s00158-019-02413-5
14. Chentanez N, Müller M (2011) Real-time eulerian water simulation using a restricted tall cell grid. In: Acm siggraph 2011 papers. ACM Digital Library, pp 1–10, https://doi.org/10.1145/1964921.1964977
15. Chollet F et al (2015) Keras. https://keras.io
16. Crisfield MA (1991) Nonlinear finite element analysis of solids and structures. Volume 1: Essentials. Wiley, New York, NY United States
17. de Borst R (2018) Finite Element Methods. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53605-6_13-1
18. Eggersmann R, Kirchdoerfer T, Reese S et al (2019) Model-free data-driven inelasticity. Comput Methods Appl Mech Eng. https://doi.org/10.1016/j.cma.2019.02.016
19. Eisen-Enosh A, Farah N, Burgansky-Eliash Z et al (2017) Evaluation of critical flicker-fusion frequency measurement methods for the investigation of visual temporal resolution. Sci Rep 7(1):15621. https://doi.org/10.1038/s41598-017-15034-z
20. Ezati M, Ghannadi B, McPhee J (2019) A review of simulation methods for human movement dynamics with emphasis on gait. Multibody Syst Dyn. https://doi.org/10.1007/s11044-019-09685-1
21. Forrester AI, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. Proc R Soc Lond A Math Phys Eng Sci 463(2088):3251–3269. https://doi.org/10.1098/rspa.2007.1900
22. Ghannadi B, Mehrabi N, Razavian RS et al (2017) Nonlinear model predictive control of an upper extremity rehabilitation robot using a two-dimensional human-robot interaction model. In: Proc. IEEE/RSJ International Conf. intelligent robots and systems (IROS), pp 502–507, https://doi.org/10.1109/IROS.2017.8202200
23. Granata KP, Wilson SE, Padua DA (2002) Gender differences in active musculoskeletal stiffness. part I.: quantification in controlled measurements of knee joint dynamics. J Electromyogr Kinesiol 12(2):119–126. https://doi.org/10.1016/S1050-6411(02)00002-0
24. Günther M, Schmitt S, Wank V (2007) High-frequency oscillations as a consequence of neglected serial damping in hill-type muscle models. Biol Cybern. https://doi.org/10.1007/s00422-007-0160-6
25. Han PH, Chen KW, Hsieh CH et al (2016) Ar-arm: Augmented visualization for guiding arm movement in the first-person perspective. In: Proc. Augmented Human International Conf., pp 1–4, https://doi.org/10.1145/2875194.2875237
26. Hawkins D, Bey M (1994) A comprehensive approach for studying muscle-tendon mechanics. J Biomech Eng. https://doi.org/10.1115/1.2895704
27. Hill AV (1938) The heat of shortening and the dynamic constants of muscle. Proc R Soc Lond B Biol Sci. https://doi.org/10.1098/rspb.1938.0050
28. Hodgins JK, Carlson DA (1996) Simulation levels of detail for real-time animation. Georgia Tech Digital Repository http://hdl.handle.net/1853/3518
29. Holzapfel GA, Linka K, Sherifova S et al (2021) Predictive constitutive modelling of arteries by deep learning. J R Soc Interface 20210411. https://doi.org/10.1098/rsif.2021.0411
30. Kässinger J, Rosin D, Dürr F et al (2022) Persival: Simulating complex 3d meshes on resource-constrained mobile ar devices using interpolation. In: Proc. IEEE International Conf. Distributed Computing Systems (ICDCS), pp 961–971, https://doi.org/10.1109/ICDCS54860.2022.00097
31. Khakhutskyy V, Hegland M (2016) Spatially-dimension-adaptive sparse grids for online learning. In: Proc. Sparse Grids and Applications-Stuttgart, Springer, pp 133–162, https://doi.org/10.1007/978-3-319-28262-6_6
32. Klotz T, Bleiler C, Röhrle O (2021) A physiology-guided classification of active-stress and active-strain approaches for

continuum-mechanical modeling of skeletal muscle tissue. Front Physiol. https://doi.org/10.3389/fphys.2021.685531
33. Kneifl J, Rosin D, Avci O (2023) Low-dimensional data-based surrogate model of a continuum-mechanical musculoskeletal system based on non-intrusive model order reduction. Arch Appl Mech. https://doi.org/10.1007/s00419-023-02458-5
34. Li P, Pei Y, Li J (2023) A comprehensive survey on design and application of autoencoder in deep learning. Appl Soft Comput 138:110176. https://doi.org/10.1016/j.asoc.2023.110176
35. Lim SK, Loo Y, Tran NT et al (2018) Doping: Generative data augmentation for unsupervised anomaly detection with gan. In: Proc. IEEE International Conf. Data Mining (ICDM), pp 1122–1127, https://doi.org/10.1109/ICDM.2018.00146
36. Lloyd JE, Stavness I, Fels S (2012) Artisynth: A fast interactive biomechanical modeling toolkit combining multibody and finite element simulation. Soft Tissue Biomech Model Comput Assisted Surgery pp 355–394. https://doi.org/10.1007/8415_2012_126
37. Maier B, Emamy N, Krämer A et al (2019) Highly parallel multiphysics simulation of muscular activation and emg. In: COUPLED VIII: Proc. International Conf. Computational Methods for Coupled Problems in Science and Engineering, CIMNE, pp 610–621, isbn:978-84-949194-5-9
38. Maier B, Göddeke D, Huber F et al (2021) Opendihu—efficient and scalable software for biophysical simulations of the neuromuscular system. J Computat Phys SSRN 4474485
39. Mendizabal A, Márquez-Neila P, Cotin S (2020) Simulation of hyperelastic materials in real-time using deep learning. Med Image Anal 101569. https://doi.org/10.1016/j.media.2019.101569
40. Meszaros-Beller L, Hammer M, Schmitt S (2023) Effect of neglecting passive spinal structures: a quantitative investigation using the forward-dynamics and inverse-dynamics musculoskeletal approach. Front Physiol 1135531. https://doi.org/10.3389/fphys.2023.1135531
41. Miotto R, Wang F, Wang S (2018) Deep learning for healthcare: review, opportunities and challenges. Brief Bioinform. https://doi.org/10.1093/bib/bbx044
42. Mooney M (1940) A theory of large elastic deformation. J Appl Phys. https://doi.org/10.1063/1.1712836
43. Neal ML, Kerckhoffs R (2010) Current progress in patient-specific modeling. Brief Bioinform 11(1):111–126. https://doi.org/10.1093/bib/bbp049
44. Ojala M, Garriga GC (2010) Permutation tests for studying classifier performance. J Mach Learn Res
45. Pappas GP, Asakawa DS, Delp SL et al (2002) Nonuniform shortening in the biceps brachii during elbow flexion. J Appl Physiol 92(6):2381–2389. https://doi.org/10.1152/japplphysiol.00843.2001
46. Platzer A, Leygue A, Stainier L et al (2021) Finite element solver for data-driven finite strain elasticity. Comput Methods Appl Mech Eng 113756. https://doi.org/10.1016/j.cma.2021.113756
47. Röhrle O (2018) Skeletal muscle modelling. Encyclopaedia for Continuum Mechanics Section Biomechanics pp 2292–2301. https://doi.org/10.1007/978-3-662-55771-6
48. Rivlin RS (1948) Large elastic deformations of isotropic materials iv. further developments of the general theory. Philos Trans R Soc Lond Ser A Math Phys Eng Sci. https://doi.org/10.1098/rsta.1948.0024
49. Röhrle O, Sprenger M, Schmitt S (2017) A two-muscle, continuum-mechanical forward simulation of the upper limb. Biomech Model Mechanobiol. https://doi.org/10.1007/s10237-016-0850-x
50. Sadeghi AH, El Mathari S, Abjigitova D et al (2020) Current and future applications of virtual, augmented, and mixed reality in cardiothoracic surgery. Annal Thoracic Surgery. https://doi.org/10.1016/j.athoracsur.2020.11.030
51. Sahrmann AS, Handsfield GG, Gizzi L et al (2024) A system for reproducible 3d ultrasound measurements of skeletal muscles. IEEE Trans Biomed Eng 71(7):2022–2032. https://doi.org/10.1109/TBME.2024.3359854
52. Sprenger M (2015) A 3D continuum-mechanical model for forward-dynamics simulations of the upper limb. Stuttgart: Institut für Mechanik (Bauwesen), Lehrstuhl für Kontinuumsbiomechanik und Mechanobiologie, isbn: 978-3-946412-01-4
53. Sudret B, Marelli S, Wiart J (2017) Surrogate models for uncertainty quantification: An overview. In: 2017 11th European conference on antennas and propagation (EUCAP), IEEE, pp 793–797, https://doi.org/10.23919/EuCAP.2017.7928679
54. Taç V, Linka K, Sahli-Costabal F et al (2023) Benchmarking physics-informed frameworks for data-driven hyperelasticity. Computat Mech pp 1–17. https://doi.org/10.1007/s00466-023-02355-2
55. Tang R, Yang XD, Bateman S et al (2015) Physio@ home: Exploring visual guidance and feedback techniques for physiotherapy exercises. In: Proc. ACM Conf. Human Factors in Computing Systems, pp 4123–4132, https://doi.org/10.1145/2702123.2702401
56. Valentin J (2019) B-splines for sparse grids: Algorithms and application to higher-dimensional optimization. https://doi.org/10.48550/arXiv.1910.05379, arXiv:1910.05379
57. Valentin J, Pflüger D (2016) Hierarchical gradient-based optimization with b-splines on sparse grids. Sparse Grids and Applications-Stuttgart 2014. Springer, pp 315–336. https://doi.org/10.1007/978-3-319-28262-6_13
58. Valentin J, Sprenger M, Pflüger D et al (2018) Gradient-based optimization with b-splines on sparse grids for solving forward-dynamics simulations of three-dimensional, continuum-mechanical musculoskeletal system models. Int J Numer Methods Biomed Eng e2965. https://doi.org/10.1002/cnm.2965
59. Van den Bogert AJ, Geijtenbeek T, Even-Zohar O et al (2013) A real-time system for biomechanical analysis of human movement and muscle function. Med Biolog Eng Comput pp 1069–1077. https://doi.org/10.1007/s11517-013-1076-z
60. Weiss JA, Gardiner JC (2001) Computational modeling of ligament mechanics. Critical Reviews in Biomedical Engineering. https://doi.org/10.1615/CritRevBiomedEng.v29.i3.20
61. Wriggers P (2013) Nichtlineare finite-element-methoden. Springer-Verlag, isbn: 978-3-540-67747-5
62. Yu X, Angerbauer K, Mohr P, et al (2020) Perspective matters: Design implications for motion guidance in mixed reality. In: Proc. IEEE international symposium mixed and augmented reality (ISMAR), pp 577–587, https://doi.org/10.1109/ISMAR50242.2020.00085
63. Zajac FE (1989) Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Critical Rev Biomed Eng pp 359–411. http://europepmc.org/abstract/MED/2676342
64. Zheng YP, Mak AF, Lue B, et al (1999) Objective assessment of limb tissue elasticity: development of a manual indentation procedure. Journal of Rehabilitation Research and Development pp 71–85

David Rosin is a doctoral researcher working on surrogate modelling of continuum-biomechanical simulations for visual real-time applications and pervasive simulation.

Johannes Kässinger is a doctoral researcher working on parallel and distributed systems in the context of heterogeneous hardware combinations and pervasive simulation.

Xingyao Yu is a doctoral researcher working on wearable interaction and virtual/augmented reality, especially with applications in motion guidance and biomechanical visualisation.

Michael Sedlmair is the leader of the research group for visualisation and virtual/augmented reality, focusing on the visualisation of complex data and interactive systems.

Okan Avci is a research group leader in the field of in-silico orthopedics, focusing on continuum-biomechanics, numerical optimisation and FE modelling, including patient-specific aspects.

Christian Bleiler is a postdoctoral researcher mainly focusing on multi-scale modelling of biological tissue and continuum-mechanical homogenisation theories.

Oliver Röhrle is professor for continuum-biomechanics and mechanobiology, focusing on multi-scale/-physics, continuum-mechanical modelling of skeletal muscles and the neuromuscular system.