Multi-fidelity surrogate modeling through hybrid machine learning for biomechanical and finite element analysis of soft tissues

Seyed Shayan Sajjadinia [a], Bruno Carpentieri [a,*], Duraisamy Shriram [b], Gerhard A. Holzapfel [c,d]

[a] *Faculty of Computer Science, Free University of Bozen-Bolzano, 39100 Bozen-Bolzano, Italy*
[b] *Mechanical Engineering Faculty, School of Engineering and Computing, American International University, Kuwait*
[c] *Institute of Biomechanics, Graz University of Technology, 8010 Graz, Austria*
[d] *Department of Structural Engineering, Norwegian University of Science and Technology, Trondheim, Norway*

ARTICLE INFO

ABSTRACT

Biomechanical simulation enables medical researchers to study complex mechano-biological conditions, although for soft tissue modeling, it may apply highly nonlinear multi-physics theories commonly implemented by expensive finite element (FE) solvers. This is a significantly time-consuming process on a regular computer and completely inefficient in urgent situations. One remedy is to first generate a dataset of the possible inputs and outputs of the solver in order to then train an efficient machine learning (ML) model, i.e., the supervised ML-based surrogate, replacing the expensive solver to speed up the simulation. But it still requires a large number of expensive numerical samples. In this regard, we propose a hybrid ML (HML) method that uses a reduced-order model defined by the simplification of the complex multi-physics equations to produce a dataset of the low-fidelity (LF) results. The surrogate then has this efficient numerical model and an ML model that should increase the fidelity of its outputs to the level of high-fidelity (HF) results. Based on our empirical tests via a group of diverse training and numerical modeling conditions, the proposed method can improve training convergence for very limited training samples. In particular, while considerable time gains comparing to the HF numerical models are observed, training of the HML models is also significantly more efficient than the purely ML-based surrogates. From this, we conclude that this non-destructive HML implementation may increase the accuracy and efficiency of surrogate modeling of soft tissues with complex multi-physics properties in small data regimes.

1. Introduction

Biomechanical simulation, especially with numerical approaches, e.g., FE methods [1], is widely used in biomedical science to analyze biological and biomechanical phenomena using engineering mechanics and biophysical principles, which have now numerous applications in, e.g., orthopedics [2–4], pathophysiology [5–7], dentistry [8–10], etc. Simulations with soft tissue models can, however, be based on complex and nonlinear multi-physics equations, which can be handled numerically at the expense of high computational costs and running times [11]. This is particularly problematic in the case of time-critical medical applications, which can be addressed by optimized algorithms, e.g., using GPU-based execution [12]. However, they may require special software and hardware settings [13] that are not always clinically available.

Alternatively, we can use the so-called surrogates, i.e., fast and possibly straightforward models that replace the expensive main models in order to overcome the computational limitations, e.g., [14–17]. In particular, a surrogate can learn soft tissue behavior with the basic idea of training on previously generated numerical results with a supervised ML method. After training, the surrogate can then be used in place of the expensive numerical model, allowing fast, and possibly real-time simulations. This method of surrogate modeling is often used for biomechanical applications, see, e.g., the simulation of prostate deformation with a feed-forward neural network (FFNN) [18], analysis of the deformation of breast and liver tissues with tree-based methods [19], assessment of the aortic aneurysm using support vector machines [20], and simulation of the passive cardiac mechanics with graph neural network (GNN) models [21]. Nevertheless, despite their appealing functionality (as discussed in detail by Phellan et al. [22]), they are still underused since they may require the generation of expensive numerical samples with lengthy training and tuning steps [23].

A possible solution is to incorporate physical principles and other relevant domain-specific knowledge into the training algorithm, e.g.,

---

* Corresponding author.
E-mail address: bruno.carpentieri@unibz.it (B. Carpentieri).

[24–26], which may improve the training convergence with few training samples. Despite the remarkable success of these attractive approaches, they are still the subject of much research as they may require craftsmanship and complex implementation considering that they typically introduce additional constraints into the training process, e.g., by penalizing the loss function of conventional learning algorithms, and in particular they may struggle with multi-physics problems and contact discontinuities of mechanical models [27–30]. Another possible solution is to directly reduce the order of the numerical model that can extract the most relevant information, which allows a simulation with considerably fewer parameters, e.g., [31,32]. The gain in simulation efficiency of such methods has led to a growing interest in different approaches of model order reduction (MOR), e.g., for surgery simulators via virtual reality [33]. But again, despite their promising performance they can have comparatively non-trivial implementation methods [34].

In this regard, the multi-fidelity MOR approaches can serve the same purpose as the hybrid methods mentioned above, but with a potentially non-destructive implementation approach. A common multi-fidelity approach is to use an LF model instead of the HF model, while an ML model is used to increase the accuracy of the generated LF results [35]. Such surrogate models have already been used in various application areas with different LF modeling techniques, see, e.g., [36–38]. But to the best of our knowledge, these methods have not been implemented through a technique of LF modeling based on the multi-physics models of soft tissues, particularly within the biomechanical context of this study.

This paper proposes a new HML framework that unifies FE methods with traditional supervised surrogates to effectively simulate complex soft tissue models in a straightforward way. It is based on an LF numerical component that only takes into account the approximate and overall behavior of the modeled tissue by simplifying the multi-physics constitutive (material) equations. Then, the accuracy of the generated LF results, via an ML model, increases to reach the HF accuracy of the expensive main model. The proposed multi-fidelity technique can be implemented by any commercially available numerical solver since it does not require low-level access to the constitutive or element code.

In short, our contributions are as follows: (i) a new multi-fidelity approach with a novel and straightforward LF modeling technique is developed to speed up the main numerical simulation with a few HF training samples; (ii) the feasibility of this HML method for surrogate modeling of soft tissue biomechanics is empirically investigated and compared to its baselines (i.e., customized GNN and FFNN surrogates) at different scales to determine possible improvements in ML training convergence and runtimes of numerical data generation.

2. Materials and methods

2.1. FE concept

FE schemes can solve differential equations by approximating the solution using piece-wise polynomials with values at specific points, i.e., integration points and nodal points. The classical FE implementation of linear systems follows the following steps [1]:

(i) Discretization of the model domain (e.g., subdivision of the materials in Euclidean space into a group of elements, i.e., FEs).
(ii) Approximating the solution parameters in all elements using interpolation functions.
(iii) Creation of a local stiffness matrix, e.g., using the Galerkin method, in order to find a linear relationship between the unknown and known values.
(iv) Assembling the stiffness matrices into a unique global stiffness matrix to define a linear system of equations.
(v) Solving the linear equations.

In practice, the FE methods may also require implementation of different iterative algorithms, e.g., for linearization, which may cause the HF multi-physics models to become extremely expensive, and therefore surrogate models can be useful.

For further clarification, an FE model is illustrated in Fig. 1. Typically, each model may have initial, boundary, and regional conditions, and they may be influenced by applied loads (or other effects on the model). These constraints are systematically added to the governing equations.

2.2. HML modeling

In this paper, the term ML, unless is not used in the context of the hybrid implementation, refers to the traditional ML-based surrogates, as shown in Fig. 2(a). For such a supervised model, an FE model first generates some informative training samples with different results at each (nodal or integration) point using the multi-physics model, as explained in the previous section. While these HF results can be viewed as the target ML vector, the input feature vector can encompass the boundary conditions, constitutive behavior, etc. In this way, the surrogate can then perform new efficient simulations, which are defined by changing the values of its inputs.

For our HML framework, we first select an LF model, which is formed by simplifying the multi-physics equations, as explained in the next section. This numerical part of reduced order, which provides efficiently a rough estimate of the tissue biomechanics, is then inserted before an ML part, which only focuses on improving the outcomes of the LF results to the level of HF results. In other words, the input features of the ML part are the components of a vector that contains the outputs computed by the LF model, as shown in Fig. 2(b).

Although this modeling method offers great flexibility in the selection of the LF model, it also requires the creation of two training datasets with different numerical methods. However, this is not a major limitation, considering that the LF model is supposed to be a simplified and faster version of the HF model with a straightforward implementation. Besides, we will empirically show that it may need, in aggregate, fewer training samples (to reach the same accuracy) and may even have faster training convergence, reducing the commonly high computational cost of the training and data generation.

2.3. MOR by simplifying multi-physics equations

The governing equations of the biomechanical systems are defined to satisfy basic equations of physical processes. In particular, the constitutive equations may correlate the stress and strain values, which can be roughly interpreted as load distributions or relative deformation effects. As a representative example, an HF multi-physics model is defined by biphasic equations, which describe the material as a combination of a fluid phase and an effective solid phase by [39,40]

$$\nabla \cdot (\boldsymbol{\sigma}_{\text{T}}) = 0, \tag{1}$$

$$\nabla \cdot (\dot{\boldsymbol{u}} - \kappa \nabla p) = 0, \tag{2}$$

where p is the fluid pressure, $\boldsymbol{\sigma}_{\text{T}}$ is the total stress, ∇ is the spatial divergence operator, $\dot{\boldsymbol{u}} = \mathrm{d}\boldsymbol{u}/\mathrm{d}t$ is the velocity of the solid phase, t is the time, and κ is the permeability (which implies the interactions between the phases). Alternatively, such a complex multi-physics model may be approximated by an LF model, e.g., via a monophasic model that cannot simulate the contribution of each phase but rather the overall responses using the LF stress $\boldsymbol{\sigma}_{\text{LF}}$ that approximates $\boldsymbol{\sigma}_{\text{T}}$ using

$$\nabla \cdot (\boldsymbol{\sigma}_{\text{LF}}) = 0. \tag{3}$$

Note that these equations have been widely used for biomechanical simulations of soft tissues. However, combinations of other similar equations can be applied depending on the definition of the physics
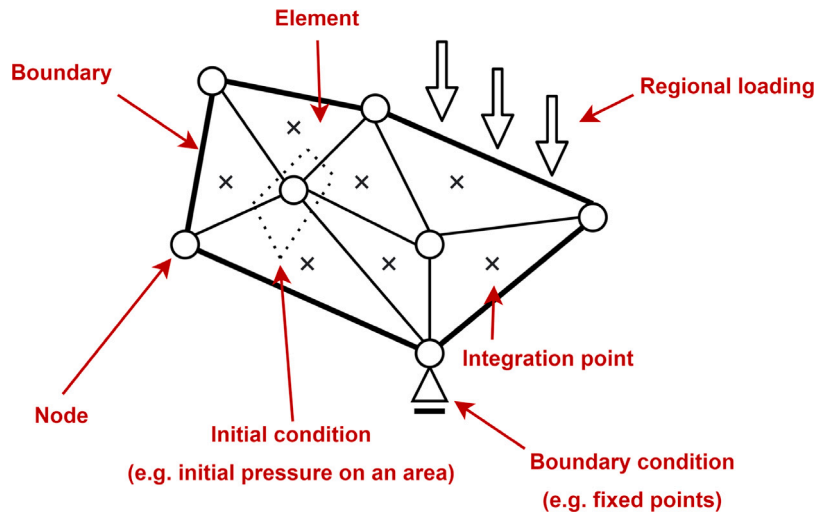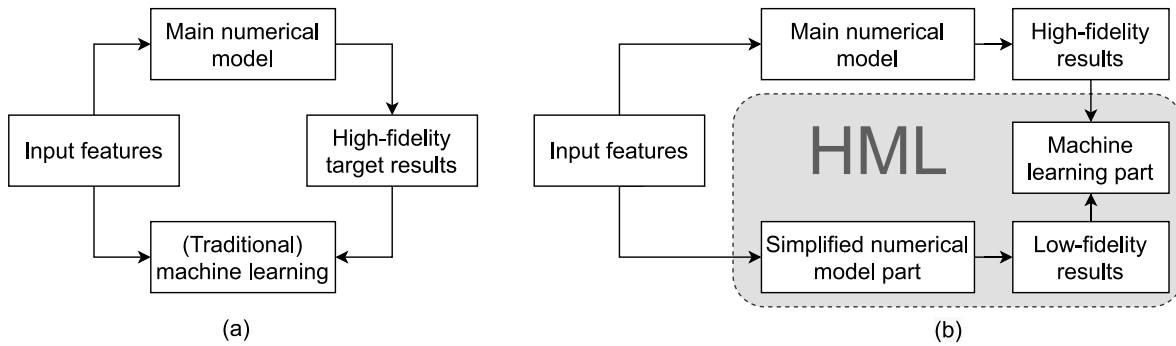
Fig. 1. Typical components of a finite element model.

Fig. 2. Scheme of a typical machine learning surrogate (**a**) and its multi-fidelity implementation (**b**). Abbreviation: HML = hybrid machine learning.

problem. In particular, FE solvers may utilize the classical consolidation theories to approximate the Eqs. (1) and (2). In addition, especially in small-scale studies, multiphasic models may be used to simulate the effects of other phases; see this survey [11] for more details on the relationships between these equations and their interpretations.

Eq. (3), which governs the LF model, although comparing to the Eqs. (1) and (2) has less accuracy, but it may also work more efficiently because it deals with fewer parameters and may not use an integration procedure with small increments that calculates the contribution of the fluid phase. In our simulation tests, in the next section, we considered two LF models of a similar type.

2.4. Simulation tests

In this section we design two physics problems for the numerical generation of the datasets using the FE methods in Abaqus [41]. The ML models are then implemented and analyzed by the established Python libraries [42–46]. The simulations are defined based on typical two-dimensional (2D) and three-dimensional (3D) soft tissue tests that might be used for applications that require iterative biomechanical simulation.

2.4.1. Model problems

The small-scale physics problem includes 2D loading tests on an artificial soft tissue, as illustrated in Fig. 3 (left). Here, a rectangular sample with dimensions $0.3 \times 1.0 \, \text{mm}^2$ was modeled using the following HF multi-physics equations for the 2D total stress [47–49]

$$\sigma_{\text{T(2D)}} = \sigma_{\text{HE(2D)}} - \Delta\pi, \tag{4}$$

where $\Delta\pi$ is the commonly simulated osmotic pressure (induced by the electrochemical nature of the tissue), and $\sigma_{\text{HE(2D)}}$ refers to the selected hyperelastic tissue response (modeling the nonlinear elasticity in the tissue). This constitutive behavior was extracted from our previous study [50], but now we disregarded the fibrillar and fluid contributions to further speed up the HF simulation. However, due to the pre-stress induced by the osmotic pressure, some of the initial states may not initiate correctly (e.g., the initial geometry may expand before the main simulation, resulting in an incorrect initial shape). Accordingly, a pre-stressing algorithm must be performed to correctly initiate the numerical simulation, and the LF model is simply defined by ignoring the pre-stressing algorithm.

Given the importance of out-of-distribution sampling [51,52] and given the simplicity of the small-scale modeling problem, we also assess the generalizability of the model using another 2D model, shown in Fig. 3 (right), where the loads are applied at different locations, and the model has also a different shape (i.e., a $0.9 \times 0.9 \, \text{mm}^2$ sample with a circular hole of 0.2 mm radius in the center).

On the other hand, the 3D simulations are carried out on a human knee cartilage model using another multi-physics model, similar to the representative example described in Section 2.3, in order to determine the effects of different body weight values on a cartilage substructure (see Fig. 4), as a typical large-scale test. For this, a 3D total stress equation is employed as follows [47,48]

$$\sigma_{\text{T(3D)}} = \sigma_{\text{HE(3D)}} - p\mathbf{I}, \tag{5}$$

where $\sigma_{\text{HE(3D)}}$ is the chosen 3D hyperelastic equation and $\mathbf{I}$ denotes the identity tensor. A simpler monophasic model for the LF modeling is also defined, this time by neglecting the fluid contribution p.
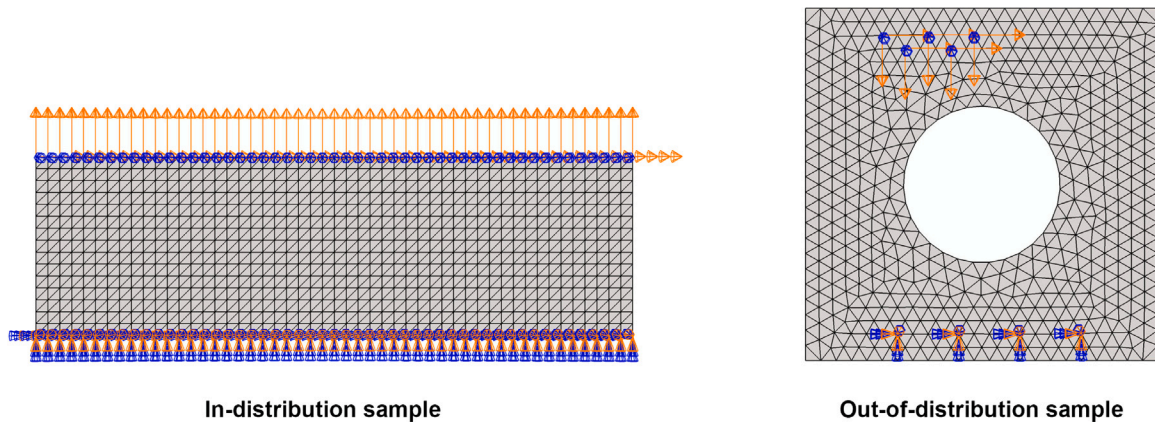
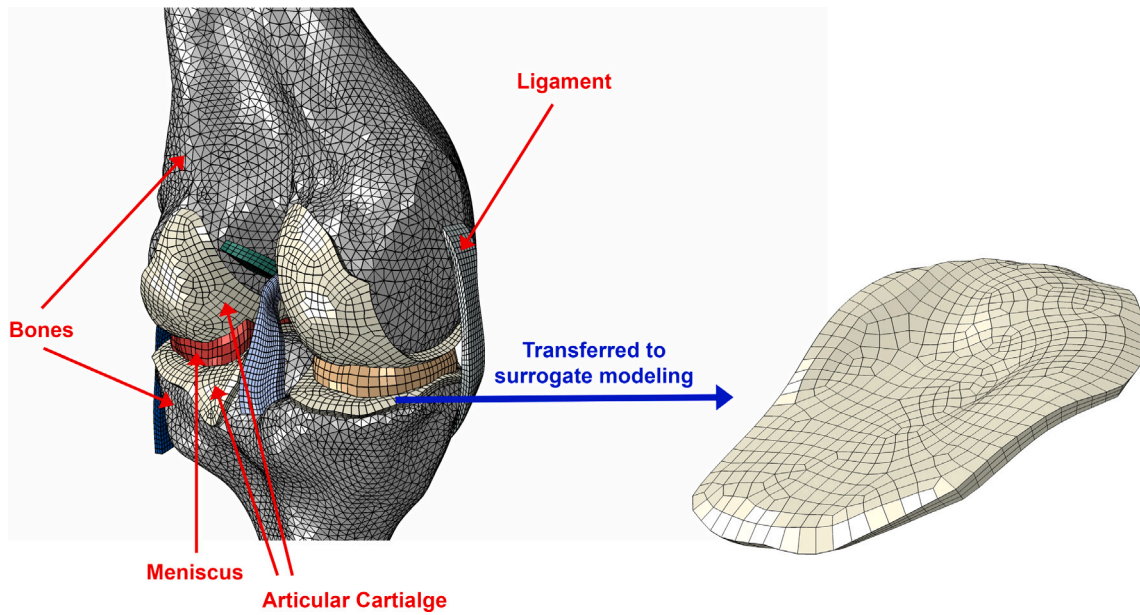**Fig. 3.** Different types of small-scale numerical samples used for surrogate modeling.

Fig. 4. Large-scale numerical modeling of a human knee and the cartilage substructure used for surrogate modeling.

2.4.2. Numerical implementation

Eq. (1) is implemented for both of the 2D LF and HF models using the following common constitutive equations [7,50,53–56]:

$$\Delta \boldsymbol{\pi} = \alpha_1 \left(\frac{1}{\det \boldsymbol{F}} \right)^{\alpha_2} \boldsymbol{I}, \tag{6}$$

$$\boldsymbol{\sigma}_{\text{HE(2D)}} = \emptyset_0^S \frac{0.61 G_m}{J} \left[-\frac{\ln J}{6} \left(3\emptyset_0^S \frac{J \ln J}{(J - \emptyset_0^S)^2} - 1 - 3\frac{J + \emptyset_0^S}{J - \emptyset_0^S} \right) \boldsymbol{I} \right.$$
$$\left. + \left(\boldsymbol{F} \boldsymbol{F}^{\text{T}} - J^{2/3} \boldsymbol{I} \right) \right], \tag{7}$$

where $\emptyset_0^S = 0.15$ is the initial solid volume fraction (and since the fluid phase is neglected, it acts here as the elastic material parameter that determines the compressibility of the tissue), $\alpha_2 = 3.22$ is another material constant, α_1 is a material parameter that also controls the pre-stress in the tissue, considered as an input value of the surrogate model, and J is the determinant of the deformation gradient $\boldsymbol{F}$ correlated to the deformation field $\boldsymbol{u}$ via [57]

$$\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}. \tag{8}$$

In these simulations, the bottom of the tissue is fully constrained, and the top is subject to displacement boundary conditions. Results

are calculated using three-node linear plane strain triangular element formulations in Abaqus. Also, as already mentioned, the HF models are specifically initiated by a pre-stressing algorithm to get accurate results. In short, the chosen pre-stressing scheme checks iteratively the updated states of the model (including the geometry and the updated material states) induced by the initial osmotic pressure, unless it reaches the experimentally observed initial states. This is implemented using a custom Fortran user material subroutine and an inverse FE method, as explained in our previous study [50].

Also in the 3D HF simulations, the interaction of the fluid and solid phases of the articular cartilage substructures is approximated by the Biot's theory in Abaqus, with the fluid phase being an inviscid water phase with the same permeability values used in previous studies for the middle zone of the cartilage model [7,58]. The solid phase is a nearly incompressible neo-Hookean model, i.e. [57,59]

$$\boldsymbol{\sigma}_{\text{HE(3D)}} = \frac{2}{J} \boldsymbol{F} \frac{\partial \Psi}{\partial \boldsymbol{C}} \boldsymbol{F}^{\text{T}}, \qquad \Psi = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1}(J - 1)^2, \tag{9}$$

where $C_{10} = 2.48\,\text{MPa}$ and $D_1 = 0.59\,\text{MPa}^{-1}$ are material constants, Ψ is the corresponding strain–energy function, and $\boldsymbol{C} = \boldsymbol{F}^{\text{T}} \boldsymbol{F}$ is the right Cauchy–Green tensor defining $\bar{I}_1$ as the first invariant of its deviatoric part. The LF models are then simply assumed to be the same as the solid hyperelastic part. The constitutive equations of the other

materials in the knee come from the former knee joint study [60], and the knee geometry is taken from the Open Knee Project [61] using magnetic resonance imaging of a cadaveric specimen from a 70-year-old woman. In addition, we use the eight-node 3D brick elements with and without the pore pressure formulations for the HF and LF articular cartilage models, respectively, while the bones use the rigid elements, and the other knee substructures are implemented by the eight-node 3D linear brick element formulation. The ligament–bone and cartilage–bone surfaces are bound, and the fluid can flow into or out of the cartilage surfaces freely. Ultimately, body weight is applied through the femur with a zero flexion–extension movement while the tibia is fully constrained.

Given the complexity of the numerical models, even if they are simplified, readers are encouraged to read the cited studies or the GitHub repository of this study for more details.

2.4.3. Datasets and evaluation

The datasets of the inputs and outputs of the 2D models contain 20 in-distribution samples, where the inputs are given by the continuous distribution (U), including two displacement boundary conditions from $U(-0.1\,\text{mm}, 0.1\,\text{mm})$ and a material parameter α_1 from $U(0.005, 0.010\,\text{MPa})$. To the input values we also add very small noises at each node to make the model more generalizable, and therefore the vector of the noisy initial coordinates of each node is considered as another input. The noise affects both the initial values and all the outputs as applied directly in the numerical models. The LF and HF outputs are the numerically calculated deformation and total stress values at each node in the LF and HF simulations, respectively. Likewise, 10 more samples are generated by the in-distribution numerical samples (with different input values) that make up the validation set used for the evaluation of the trained model. We also define a test set of 10 samples (taking into account the limited samples available) with again different input values, but now we use the out-of-distribution numerical model to assess the surrogate generalizability.

A total of 100 samples are created for the 3D surrogate modeling. The input to the HF models consists of the weight load, from $U(-900\,\text{N}, -400\,\text{N})$, and the outputs are the calculated normal stress and strain values at each nodal point (extrapolated from the integration points), while the LF outputs are the nodal deformation field. The dataset is shuffled and divided into the validation, training, and test sets while we set the validation size to a third size of the training set, and the remaining samples form the test set. In the results we specify the exact number of training samples, which should be very small, since generating such data is expensive and in most cases limited.

Note that most of the results of the 2D samples are recorded in a similar range from 10^{-1} to 10^{-3} (in terms of their units and their absolute values). Therefore, the unscaled mean squared error (MSE) metric based on all nodal outputs of each evaluation sample is used for evaluation. In contrary, for the 3D surrogates, since the inputs and outputs have very different scales, they are all normalized separately based on the training samples to aid in training and evaluation. While this scaling provides a fair comparison between different output types, the MSE metric can give very misleading results (since it may average the outlier nodes with the other scaled nodal outputs), hence the pointwise MSE (PMSE) metric is used, since it averages these scaled nodal outputs at each node separately. The exact equations of the MSE and PMSE metrics are explained and compared in our previous study [62].

2.4.4. Training

The 2D surrogate was developed based on the open-source Mesh-GraphNets framework [63], a GNN architecture consisting of a set of connected FFNNs divided into three subsets of the encoder, processor, and decoder [64,65]. In this way, the model is trained on each node and its corresponding edges, hopefully to better learn the local behavior, in particular by transferring geometrical data to the bidirectional edge

sets, allowing for spatial equivariance and more generalizable results. In short, the core GNN model first encodes the input features of the node and edge sets into their latent vectors to be then processed by identical and sequential message-passing blocks, as suggested by Sanchez-Gonzalez et al. [66]. The outputs of the latent vectors are also reinforced by the residual connections [65], and the final latent vectors are decoded into the desired outputs of the surrogate models.

Here we modify the original MeshGraphNets algorithm by disabling the domain-specific settings, such as the separate world and mesh edges definitions (since no mechanical contact is defined in the 2D simulations). Most importantly, their training noise strategy is not applicable to the selected output types of this study and is therefore replaced by our noise generation method described in Section 2.4.3. Also, we edited the pre-processing functions and the trained model evaluators since they were developed for dynamic systems with different snapshots of the results, while all the simulations of this study consider only the end conditions as output results. Using this customized model, the ML and HML implementations are empirically compared.

Accordingly, we define 2D model 1 in such a way that it can be efficiently trained with different numbers of input samples, using the L_2 loss of the nodal results with a maximum number of 1 000 steps and a learning rate of 10^{-3} while the first 100 steps are reserved for initiating the online normalizer [65]. Each of the constituting FFNNs has three hidden layers while we insert three message-passing blocks with a latent size of 40, which corresponds to the output size of the encoders and the input size of the decoders. The edge sets are generated by the vectors of relative distances of connected nodes and their norms; thus, the input shape of their encoders is of size 3. For the HML model, the outputs of the LF model with respect to each node in the sample yields the corresponding node set; hence the input size of its encoder is 6. This size for the ML model is 5 since each graph's node is formed by the input values of the numerical model (sent to each node), excluding the geometrical features, but containing also a one-hot vector identifying the fully constrained nodes. The output size of the decoders of all surrogates agrees with the vector of the HF prediction (i.e., 6). While the batch size is set equal to the number of all nodes of each in-distribution sample (i.e., 816), the other hyperparameters are the same as the original code.

We also empirically compare the 3D models trained by commonly used FFNNs [22]. To ensure fast convergence, the Adam optimizer [67] is used while keeping the numbers of neurons and layers small. After manually testing various values of the hyperparameters for surrogate modeling with efficient training, 3D model 1 is defined with four hidden layers each with 940 units with the ReLU activation functions [68,69] followed by a normalization layer [70], and the output dimension is equal to the total number of nodal results in each HF sample (i.e., 94 032). While the input dimension for the ML model is 1, for the ML part of the HML model, this size is set to the total number of nodal results in each LF sample (i.e., 8 364). The test is implemented on only a few training samples with a learning rate of 10^{-4} and the other training settings are set to the library defaults.

Additionally, in the next section, more models are defined and trained to investigate the effects of the hyperparameters, and the numbers of training samples.

3. Results and discussion

The simulation tests were carried out by different LF and HF models on an Intel Core i5 CPU and 8 GB RAM. Runtime reductions, shown in Fig. 5 (and detailed in Table S1 in the Supplementary material), can justify the use of LF models instead of the HF training samples. Changing the rate of applied loading of the 3D simulation resulted in more diverse runtimes. In practice, however, the difference could be even more significant since the HF simulations are simplified here to speed up surrogate modeling.
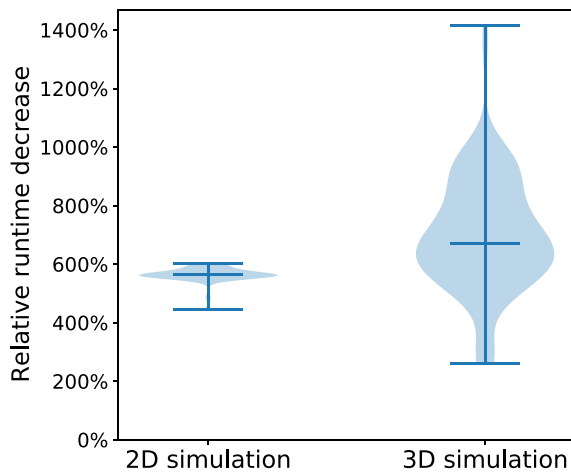
Fig. 5. Violin plot [71] of the relative runtime decrease of high-fidelity simulations in relation to the corresponding low-fidelity simulations. The data is measured by dividing the runtime decreases by the corresponding runtimes of the low-fidelity simulations.

We have disregarded the ML inference runtimes, since they were real-time and most of the computational costs of such surrogates were then caused by numerical data generation. In the next two subsections we compare HML and ML surrogates for different simulations to verify the advantages of the HML models with limited HF samples. Having two independent multi-physics tests with different surrogate modeling strategies allowed us to evaluate the functionality of the models with different training conditions. We end this section with a discussion of the limitations of the proposed method.

3.1. 2D simulation

Fig. 6 shows the MSE evaluation values for both the in-distribution and out-of-distribution groups of 2D samples. The datasets are intentionally kept small and not pre-processed by any data curation technique [72], because in practical cases of soft tissue modeling such datasets may be too small for any exclusion of numerical samples [73]. Under these circumstances, the HML models have significantly outperformed their ML counterparts, as most of the recorded errors are around 10^{-4} or less, with the exception for the error of the HML model with four training samples (but even this error is smaller than all the computed errors of the ML models). We also repeated the experiment with an alternative model (i.e., 2D model 2) with less efficient but more training parameters, adding three more message-passing steps as the key hyperparameter [63] while considering that the learning rate is reduced to 10^{-4}, the model is trained with eight times more training steps. The results are not changed considerably, demonstrating that the HML models are again more precise and generalizable.

Besides, because the HML model works by mapping the input features into another feature representation for a subsequent data-driven part, it can be compared to feature-based transfer learning approaches, in which the source features are transformed into a new feature representation to transfer some of the learned knowledge of another domain [74]; but instead of relying on the feature transformation strategies commonly used in data-driven models, e.g., [75–77], here, an LF physics-based model transforms the input features into more informative features. Such an analogy to transfer learning can explain the reasons for the observed performance improvement of the HML implementation on these efficient learning methods and limited training data.

In order to also examine the influence of the accuracy of the LF modeling on the HML modeling, we reimplemented the 2D tests via the same hybrid algorithm, but this time by entering the HF results (instead of the LF results). As shown in Table S2 in the Supplementary

material, this change resulted in relatively similar errors, since our hybrid model had already been well-trained with the LF inputs (despite the stochastic nature of the training algorithm). This might prove that the inaccuracy in the LF modeling, which has also been documented in the other relevant studies [50,78,79], did not prevent it from extracting informative data to be transferred to the downstream ML component.

3.2. 3D Simulation

Fig. 7 shows the recorded MSE loss values during training (whereas the converged values are also compared in Table S3 in the Supplementary material) and reveals that 3D model 1 with the HML implementation could outperform its ML version. 3D model 2, trained with the same number of samples (i.e., three training samples and a single validation sample for regularization) but with ELU [80] as the activation function, had non-significant impact on the results, and the HML surrogate still recorded minor errors. This is especially important as only 100 epochs were used, which makes training for surrogate modeling very fast and efficient. To further investigate the effects of the hybrid implementation, 3D model 3 was defined again on the same numbers of training and validation samples, where we increased and decreased, respectively, the number of epochs and neurons (in the hidden layers) by an order of magnitude, and we used the ReLU activation function and the early stopping hyperparameter of 100. In this way, by reducing the number of training parameters and the learning rate, we still kept training efficient over such a high number of epochs. Although these changes had a negative effect on the HML model (due to a jump in the loss of validation after about 30 epochs), it still worked better than the ML surrogate.

For the last model, the numbers of training and validation samples tripled while we used the hyperparameters similar to 3D model 1, but with a maximum of 1 000 training epochs and the same early stopping hyperparameter of 3D model 3. This made 3D model 4 considerably more expensive, since with more data, parameters and training epochs, the training and the numerical data generation became very time-consuming. The results show that the ML and HML models gave relatively similar converged losses, but still the HML surrogate converged much faster since it took about 10 epochs to record a validation loss value of 0.1, i.e. about 75 times less than the number of epochs that the ML surrogate required. Taken together, these results show that the HML method can improve both convergence and performance.

In order to clarify the significance of the observed differences in performance, Fig. 8 compares the PMSE contours that were mapped for the above models on the reference configuration of the articular cartilage substructure. 3D models 1 to 3 with the ML implementation worked poorly in all points, while their HML versions had a significantly high level of accuracy with only few outliers in pointwise errors (recognizable by small red zones). 3D model 4 showed a similar accuracy, with a negligible difference with the accuracy of the HML surrogate. However, it seems that this inefficient model recorded more pointwise outliers and therefore other more efficient HML models could be preferred, which in turn shows the advantages of HML surrogate modeling, especially for very small data regime.

3.3. Limitations and future work

This study has several limitations to report. First, the HF numerical models were simplified in order to obtain the datasets efficiently; but this is not a significant limitation, as the simulations were designed to efficiently demonstrate the importance of the HML method in relevant biomechanical simulations and are not used as validated models with a proven clinical application. Second, in our empirical tests we assumed that the training stages should not be considerably more expensive than the numerical data generation to motivate the use of these surrogates, and therefore we avoided expensive surrogate modeling settings. Although this might limit our results to our modeling conditions, such
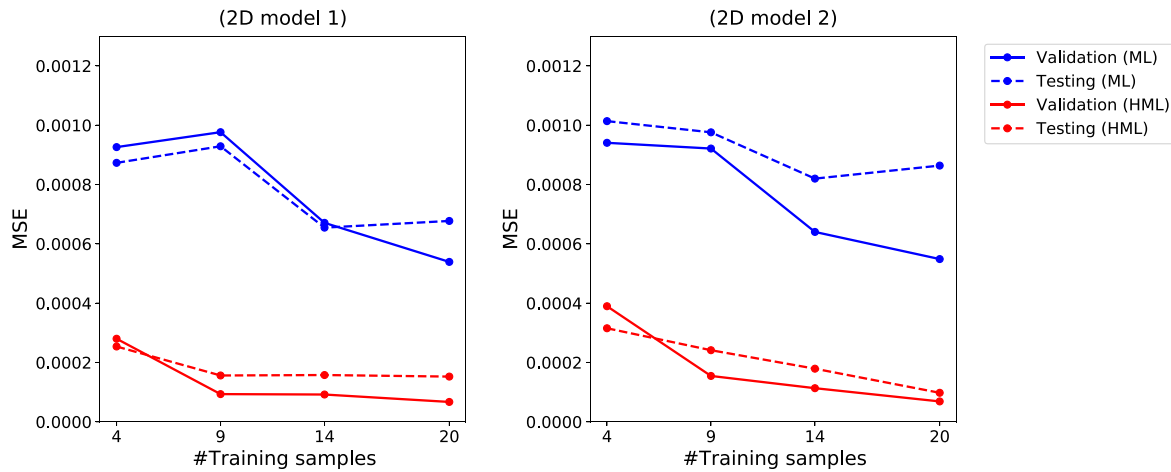
Fig. 6. Evaluation errors on validation and test sets in relation to the number of training samples in two-dimensional simulations. Abbreviations: MSE = mean squared error; ML = machine learning; HML = hybrid machine learning.
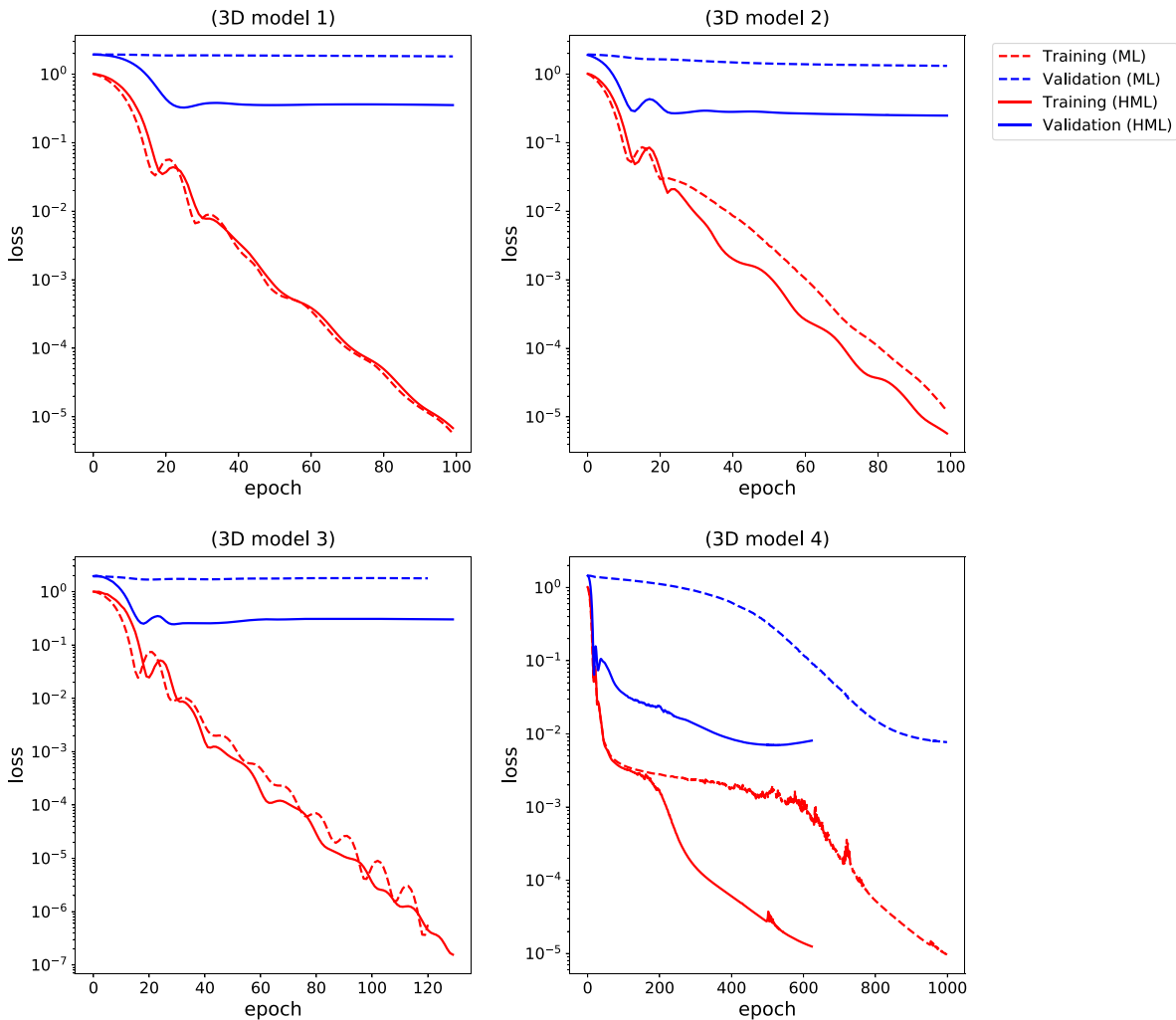

Fig. 7. Recorded loss values versus number of epochs for surrogates of three-dimensional simulations. Note that some models converged earlier due to early stopping regularization. Abbreviations: ML = machine learning; HML = hybrid machine learning.

assumptions are usually inevitable for surrogate models implemented on regular computers.

Our experiments included some (but typical) types of the biomechanical simulations to evaluate the role our hybrid method can play.

Admittedly, if more different types of simulations had been used, we would have obtained more generalizable results. However, these HF simulations may have non-trivial implementations with private datasets and are therefore left to the future.
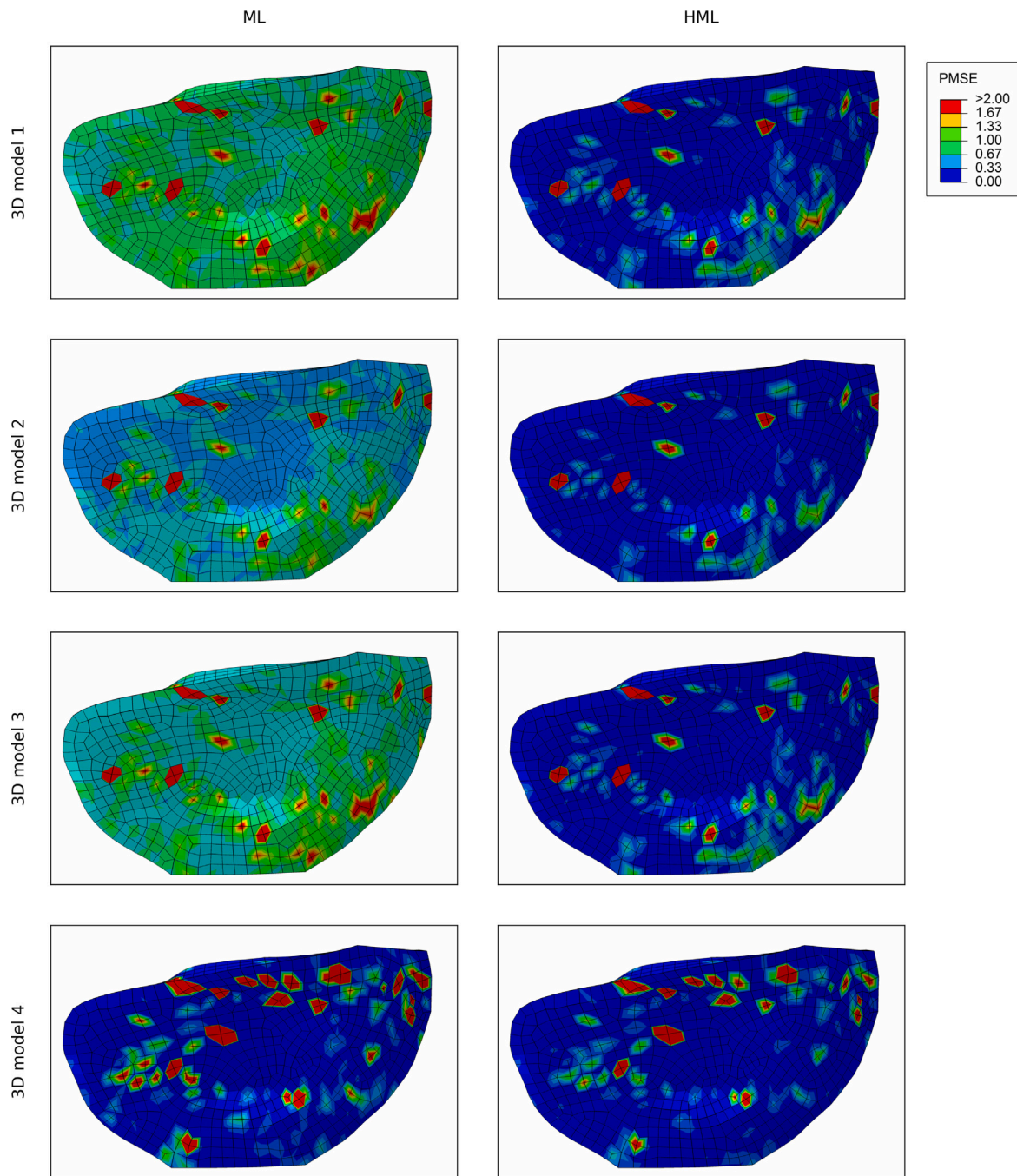
Fig. 8. Contours of the pointwise mean squared error for all three-dimensional surrogate models with pure machine learning and hybrid implementations. Abbreviations: ML = machine learning; HML = hybrid machine learning, PMSE = pointwise mean squared error.

4. Conclusions

This study presented an HML method for multi-fidelity surrogate modeling of the biomechanical and FE simulation of soft tissues. Since the LF model is only formed by simplifying the multi-physics equations, such an MOR technique is comparatively straightforward and non-destructive. Through extensive empirical comparisons, we also conclude that this hybrid paradigm can improve training convergence and increase the performance of surrogate modeling with an extremely limited number of training samples, which can address the limitation of current ML-based surrogates, which typically need a large number of expensive HF samples.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Research data

The research data is shared at github.com/shayansss/hml.

Acknowledgments

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compbiomed.2022.105699.

References

[1] T. Belytschko, W.K. Liu, B. Moran, Nonlinear Finite Elements for Continua and Structures, John Wiley & Sons, Chichester, 2000.

[2] I.S. Mohd Moideen, C.T. Lim, R.C. Yeow, D.Y. Chong, Polka dot cementless talar component in enhancing total ankle replacement fixation: A parametric study using the finite element analysis approach, Comput. Biol. Med. 141 (2022) 105142, http://dx.doi.org/10.1016/j.compbiomed.2021.105142.

[3] S.S. Sajjadinia, M. Haghpanahi, A parametric study on the mechanical role of fibrillar rotations in an articular cartilage finite element model, Sci. Iran. 28 (2) (2021) 830–836, http://dx.doi.org/10.24200/sci.2020.51785.2362.

[4] Y. Peng, D.W.-C. Wong, T.L.-W. Chen, Y. Wang, G. Zhang, F. Yan, M. Zhang, Influence of arch support heights on the internal foot mechanics of flatfoot during walking: A muscle-driven finite element analysis, Comput. Biol. Med. 132 (2021) 104355, http://dx.doi.org/10.1016/j.compbiomed.2021.104355.

[5] M. Moayedi, A. Arshi, M. Salehi, M. Akrami, R. Naemi, Associations between changes in loading pattern, deformity, and internal stresses at the foot with hammer toe during walking; A finite element approach, Comput. Biol. Med. 135 (2021) 104598, http://dx.doi.org/10.1016/j.compbiomed.2021.104598.

[6] L. Zhang, S.K. Choi, T. Xie, P. Jiang, J. Hu, J. Koo, Multi-fidelity surrogate model-assisted fatigue analysis of welded joints, Struct. Multidiscip. Optim. 63 (6) (2021) 2771–2787, http://dx.doi.org/10.1007/s00158-020-02840-9.

[7] S.S. Sajjadinia, M. Haghpanahi, M. Razi, Computational simulation of the multiphasic degeneration of the bone-cartilage unit during osteoarthritis via indentation and unconfined compression tests, Proc. Inst. Mech. Eng. H 233 (9) (2019) 871–882, http://dx.doi.org/10.1177/0954411919854011.

[8] A. Chakraborty, P. Datta, S. Majumder, S.C. Mondal, A. Roychowdhury, Finite element and experimental analysis to select patient's bone condition specific porous dental implant, fabricated using additive manufacturing, Comput. Biol. Med. 124 (2020) 103839, http://dx.doi.org/10.1016/j.compbiomed.2020.103839.

[9] L.-R. Chang, Y.-P. Hou, T.-S. Lin, Is perpendicular double two-hole plates fixation superior to single four-hole plate fixation to treat mandibular symphysis fracture?—A finite element study, Appl. Sci. 11 (18) (2021) http://dx.doi.org/10.3390/app11188629.

[10] G. Chen, W. Fan, S. Mishra, A. El-Atem, M. Schuetz, Y. Xiao, Tooth fracture risk analysis based on a new finite element dental structure models using micro-CT data, Comput. Biol. Med. 42 (10) (2012) 957–963, http://dx.doi.org/10.1016/j.compbiomed.2012.07.006.

[11] M. Freutel, H. Schmidt, L. Dürselen, A. Ignatius, F. Galbusera, Finite element modeling of soft tissues: material models, tissue interaction and challenges, Clin. Biomech. 29 (4) (2014) 363–372, http://dx.doi.org/10.1016/j.clinbiomech.2014.01.006.

[12] S.F. Johnsen, Z.A. Taylor, M.J. Clarkson, J. Hipwell, M. Modat, B. Eiben, L. Han, Y. Hu, T. Mertzanidou, D.J. Hawkes, et al., NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics, Int. J. Comput. Assist. Radiol. Surg. 10 (7) (2015) 1077–1095, http://dx.doi.org/10.1007/s11548-014-1118-5.

[13] D. Marinkovic, M. Zehn, Survey of finite element method-based real-time simulations, Appl. Sci. 9 (14) (2019) http://dx.doi.org/10.3390/app9142775.

[14] F. Martínez-Martínez, M.J. Rupérez-Moreno, M. Martínez-Sober, J.A. Solves-Llorens, D. Lorente, A.J. Serrano-López, S. Martínez-Sanchis, C. Monserrat, J.D. Martín-Guerrero, A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time, Comput. Biol. Med. 90 (2017) 116–124, http://dx.doi.org/10.1016/j.compbiomed.2017.09.019.

[15] L. Liang, M. Liu, C. Martin, W. Sun, A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis, J. R. Soc. Interface 15 (138) (2018) http://dx.doi.org/10.1098/rsif.2017.0844.

[16] X. Liu, C.E. Athanasiou, N.P. Padture, B.W. Sheldon, H. Gao, A machine learning approach to fracture mechanics problems, Acta Mater. 190 (2020) 105–112, http://dx.doi.org/10.1016/j.actamat.2020.03.016.

[17] L. Cai, L. Ren, Y. Wang, W. Xie, G. Zhu, H. Gao, Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium, Royal Soc. Open Sci. 8 (1) (2021) 201121, http://dx.doi.org/10.1098/rsos.201121.

[18] A. Jahya, M. Herink, S. Misra, A framework for predicting three-dimensional prostate deformation in real time, Int. J. Med. Robot. Comput. Assist. Surg. 9 (4) (2013) 211–215, http://dx.doi.org/10.1002/rcs.1493.

[19] J.D. Martin-Guerrero, M.J. Ruperez-Moreno, F. Martinez-Martinez, D. Lorente-Garrido, A.J. Serrano-Lopez, C. Monserrat, S. Martinez-Sanchis, M. Martinez-Sober, Machine learning for modeling the biomechanical behavior of human soft tissue, in: IEEE 16th International Conference on Data Mining Workshops, ICDMW, 2016, pp. 247–253, http://dx.doi.org/10.1109/ICDMW.2016.0042.

[20] L. Liang, M. Liu, C. Martin, J.A. Elefteriades, W. Sun, A machine learning approach to investigate the relationship between shape features and numerically predicted risk of ascending aortic aneurysm, Biomech. Model. Mechanobiol. 16 (5) (2017) 1519–1533, http://dx.doi.org/10.1007/s10237-017-0903-9.

[21] D. Dalton, A. Lazarus, A. Rabbani, H. Gao, D. Husmeier, Graph neural network emulation of cardiac mechanics, in: Proceedings of the 3rd International Conference on Statistics: Theory and Applications, ICSTA'21, 2021, p. 127, http://dx.doi.org/10.11159/icsta21.127.

[22] R. Phellan, B. Hachem, J. Clin, J.M. Mac-Thiong, L. Duong, Real-time biomechanics using the finite element method and machine learning: Review and perspective, Med. Phys. 48 (1) (2021) 7–18, http://dx.doi.org/10.1002/mp.14602.

[23] M. Frank, D. Drikakis, V. Charissis, Machine-learning methods for computational science and engineering, Computation 8 (1) (2020) 15, http://dx.doi.org/10.3390/computation8010015.

[24] C. Hoerig, J. Ghaboussi, M.F. Insana, Physics-guided machine learning for 3-D quantitative quasi-static elasticity imaging, Phys. Med. Biol. 65 (6) (2020) 065011, http://dx.doi.org/10.1088/1361-6560/ab7505.

[25] K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H.A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, n. Prabhat, Physics-informed machine learning: Case studies for weather and climate modelling, Phil. Trans. R. Soc. A 379 (2194) (2021) 20200093, http://dx.doi.org/10.1098/rsta.2020.0093.

[26] K. Linka, M. Hillgärtner, K.P. Abdolazizi, R.C. Aydin, M. Itskov, C.J. Cyron, Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning, J. Comput. Phys. 429 (2021) 110010, http://dx.doi.org/10.1016/j.jcp.2020.110010.

[27] O. Fuks, H.A. Tchelepi, Limitations of physics informed machine learning for nonlinear two-phase transport in porous media, J. Mach. Learn. Model. Comput. 1 (1) (2020) 19–37, http://dx.doi.org/10.1615/JMachLearnModelComput.2020033905.

[28] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440, http://dx.doi.org/10.1038/s42254-021-00314-5.

[29] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, Acta Mech. Sinica (2022) http://dx.doi.org/10.1007/s10409-021-01148-1.

[30] E.J.R. Coutinho, M. Dall'Aqua, L. McClenny, M. Zhong, U. Braga-Neto, E. Gildin, Physics-informed neural networks with adaptive localized artificial viscosity, 2022, http://dx.doi.org/10.48550/ARXIV.2203.08802, arXiv.

[31] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta, Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models, Comput. Methods Programs Biomed. 105 (1) (2012) 1–12, http://dx.doi.org/10.1016/j.cmpb.2010.06.012.

[32] O.J. Pellicer-Valero, M.J. Rupérez, S. Martínez-Sanchis, J.D. Martín-Guerrero, Real-time biomechanical modeling of the liver using machine learning models trained on finite element method simulations, Expert Syst. Appl. 143 (2020) 113083, http://dx.doi.org/10.1016/j.eswa.2019.113083.

[33] E. Cueto, F. Chinesta, Real time simulation for computational surgery: A review, Adv. Model. Simul. Eng. Sci. 1 (1) (2014) 1–18, http://dx.doi.org/10.1186/2213-7467-1-11.

[34] N. Lauzeral, D. Borzacchiello, M. Kugler, D. George, Y. Rémond, A. Hostettler, F. Chinesta, A model order reduction approach to create patient-specific mechanical models of human liver in computational medicine applications, Comput. Methods Programs Biomed. 170 (2019) 95–106, http://dx.doi.org/10.1016/j.cmpb.2019.01.003.

[35] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Rev. 60 (3) (2018) 550–591, http://dx.doi.org/10.1137/16M1082469.

[36] Q. Zhang, T. Chon, Y. Zhang, J.S. Baker, Y. Gu, Finite element analysis of the lumbar spine in adolescent idiopathic scoliosis subjected to different loads, Comput. Biol. Med. 136 (2021) 104745, http://dx.doi.org/10.1016/j.compbiomed.2021.104745.

[37] Z. Jiang, J. Choi, S. Baek, Machine learning approaches to surrogate multifidelity growth and remodeling models for efficient abdominal aortic aneurysmal applications, Comput. Biol. Med. 133 (2021) 104394, http://dx.doi.org/10.1016/j.compbiomed.2021.104394.

[38] Q. Zhou, P. Jiang, X. Shao, J. Hu, L. Cao, L. Wan, A variable fidelity information fusion method based on radial basis function, Adv. Eng. Inf. 32 (2017) 26–39, http://dx.doi.org/10.1016/j.aei.2016.12.005.

[39] M.H. Holmes, V.C. Mow, The nonlinear characteristics of soft gels and hydrated connective tissues in ultrafiltration, J. Biomech. 23 (11) (1990) 1145–1156, http://dx.doi.org/10.1016/0021-9290(90)90007-P.

[40] H. Guo, S.A. Maher, P.A. Torzilli, A biphasic multiscale study of the mechanical microenvironment of chondrocytes within articular cartilage under unconfined compression, J. Biomech. 47 (11) (2014) 2721–2729, http://dx.doi.org/10.1016/j.jbiomech.2014.05.001.

[41] Abaqus: Dassault Systèmes Simulia Corp., providence, RI, USA, 2021, http://www.simulia.com (retrieved on 10/01/22).

[42] J.D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. 9 (3) (2007) 90–95, http://dx.doi.org/10.1109/MCSE.2007.55.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830.

[44] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., TensorFlow: large-scale machine learning on heterogeneous systems, 2015, https://www.tensorflow.org/ (retrieved on 10/01/22).

[45] M. Andrychowicz, M. Denil, S. Gómez, M.W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. de Freitas, Learning to learn by gradient descent by gradient descent, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016, http://dx.doi.org/10.48550/ARXIV.1606.04474.

[46] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, et al., Array programming with NumPy, Nature 585 (7825) (2020) 357–362.

[47] V.C. Mow, S.C. Kuei, W.M. Lai, C.G. Armstrong, Biphasic creep and stress relaxation of articular cartilage in compression: Theory and experiments, J. Biomech. Eng. 102 (1) (1980) 73–84, http://dx.doi.org/10.1115/1.3138202.

[48] W.M. Lai, J.S. Hou, V.C. Mow, A triphasic theory for the swelling and deformation behaviors of articular cartilage, J. Biomech. Eng. 113 (3) (1991) 245–258, http://dx.doi.org/10.1115/1.2894880.

[49] W. Wilson, C. van Donkelaar, B. van Rietbergen, K. Ito, R. Huiskes, Stresses in the local collagen network of articular cartilage: A poroviscoelastic fibril-reinforced finite element study, J. Biomech. 37 (3) (2004) 357–366, http://dx.doi.org/10.1016/S0021-9290(03)00267-7.

[50] S.S. Sajjadinia, B. Carpentieri, G.A. Holzapfel, A backward pre-stressing algorithm for efficient finite element implementation of in vivo material and geometrical parameters into fibril-reinforced mixture models of articular cartilage, J. Mech. Behav. Biomed. Mater. 114 (2021) 104203, http://dx.doi.org/10.1016/j.jmbbm.2020.104203.

[51] J. Chen, L. Qian, T. Urakov, W. Gu, L. Liang, Adversarial robustness study of convolutional neural network for lumbar disk shape reconstruction from MR images, in: I. Išgum, B.A. Landman (Eds.), Medical Imaging 2021: Image Processing, Vol. 11596, SPIE, International Society for Optics and Photonics, 2021, pp. 306–318, http://dx.doi.org/10.1117/12.2580852.

[52] J.P. Cohen, P. Bertin, V. Frappier, Chester: A web delivered locally computed chest X-ray disease prediction system, 2019, http://dx.doi.org/10.48550/ARXIV.1901.11210, arXiv.

[53] M.D. Buschmann, A.J. Grodzinsky, A molecular model of proteoglycan-associated electrostatic forces in cartilage mechanics, J. Biomech. Eng. 117 (2) (1995) 179, http://dx.doi.org/10.1115/1.2796000.

[54] W. Wilson, J.M. Huyghe, C.C. Van Donkelaar, Depth-dependent compressive equilibrium properties of articular cartilage explained by its composition, Biomech. Model. Mechanobiol. 6 (1–2) (2007) 43–53, http://dx.doi.org/10.1007/s10237-006-0044-z.

[55] S.R. Oungoulian, S.S. Chen, A. Davol, R.L. Sah, S.M. Klisch, Extended two compartmental swelling stress model and isotropic cauchy stress equation for articular cartilage proteoglycans, in: ASME 2007 Summer Bioengineering Conference, in: Summer Bioengineering Conference, 2007, pp. 847–848, http://dx.doi.org/10.1115/SBC2007-175327.

[56] M.E. Stender, C.B. Raub, K.A. Yamauchi, R. Shirazi, P. Vena, R.L. Sah, S.J. Hazelwood, S.M. Klisch, Integrating qPLM and biomechanical test data with an anisotropic fiber distribution model and predictions of TGF-β 1 and IGF-1 regulation of articular cartilage fiber modulus, Biomech. Model. Mechanobiol. 12 (6) (2012) 1073–1088, http://dx.doi.org/10.1007/s10237-012-0463-y.

[57] G.A. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering, John Wiley & Sons, Chichester, 2000.

[58] M.E. Stender, R.A. Regueiro, V.L. Ferguson, A poroelastic finite element model of the bone–cartilage unit to determine the effects of changes in permeability with osteoarthritis, Comput. Methods Biomech. Biomed. Eng. 20 (3) (2017) 319–331, http://dx.doi.org/10.1080/10255842.2016.1233326.

[59] R. Willing, E.A. Lalone, G.J.W. King, J.A. Johnson, Comparing two constitutive material models of cartilage for hemiarthroplasty articular contact mechanics using computational analyses, in: Meeting of the Orthopaedic Research Society, Poster No. 1249, 2012.

[60] V.B. Shim, T.F. Besier, D.G. Lloyd, K. Mithraratne, J.F. Fernandez, The influence and biomechanical role of cartilage split line pattern on tibiofemoral cartilage stress distribution during the stance phase of gait, Biomech. Model. Mechanobiol. 15 (1) (2016) 195–204, http://dx.doi.org/10.1007/s10237-015-0668-y.

[61] A. Erdemir, Open knee: Open source modeling and simulation in knee biomechanics, J. Knee Surg. 29 (2) (2014) 107–116, http://dx.doi.org/10.1055/s-0035-1564600.

[62] S.S. Sajjadinia, B. Carpentieri, G.A. Holzapfel, A pointwise evaluation metric to visualize errors in machine learning surrogate models, in: A.J. Tallón-Ballesteros (Ed.), Proceedings of CECNet 2021, in: Frontiers in Artificial Intelligence and Applications, vol. 345, IOS Press, 2021, pp. 26–34, doi: 10.3233/faia210386.

[63] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P.W. Battaglia, Learning mesh-based simulation with graph networks, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, 2021, http://dx.doi.org/10.48550/ARXIV.2010.03409.

[64] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, R. Pascanu, Relational inductive biases, deep learning, and graph networks, 2018, http://dx.doi.org/10.48550/ARXIV.1806.01261, arXiv.

[65] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 8459–8468, http://dx.doi.org/10.48550/ARXIV.2002.09405.

[66] A. Sanchez-Gonzalez, N. Heess, J.T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, P. Battaglia, Graph networks as learnable physics engines for inference and control, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4470–4479, http://dx.doi.org/10.48550/ARXIV.1806.01242.

[67] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015, http://dx.doi.org/10.48550/ARXIV.1412.6980.

[68] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, in: ICML, vol. 10, Omni Press, Madison, WI, USA, 2010, pp. 807–814.

[69] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biol. Cybernet. 36 (4) (1980) 193–202, http://dx.doi.org/10.1007/BF00344251.

[70] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, vol. 37, PMLR, Lille, France, 2015, pp. 448–456, http://dx.doi.org/10.48550/ARXIV.1502.03167.

[71] J.L. Hintze, R.D. Nelson, Violin plots: A box plot-density trace synergism, Amer. Statist. 52 (2) (1998) 181–184, http://dx.doi.org/10.1080/00031305.1998.10480559.

[72] A. Lyngdoh, 10 - WHat we leave behind: The future of data curation, in: D. Baker, W. Evans (Eds.), Trends, Discovery, and People in the Digital Age, in: Chandos Digital Information Review, Chandos Publishing, 2013, pp. 153–165, http://dx.doi.org/10.1016/B978-1-84334-723-1.50010-3.

[73] G.R. Liu, FEA-AI And AI-AI: Two-way deepnets for real-time computations for both forward and inverse mechanics problems, Int. J. Comput. Methods 16 (08) (2019) 1950045, http://dx.doi.org/10.1142/S0219876219500452.

[74] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2021) 43–76, http://dx.doi.org/10.1109/JPROC.2020.3004555.

[75] W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Co-clustering based classification for out-of-domain documents, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD, vol. 07, Association for Computing Machinery, New York, NY, USA, 2007, pp. 210–219, http://dx.doi.org/10.1145/1281192.1281218.

[76] H. Daumé III, Frustratingly easy domain adaptation, in: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 256–263, http://dx.doi.org/10.48550/ARXIV.0907.1815.

[77] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210, http://dx.doi.org/10.1109/TNN.2010.2091281.

[78] J. Bols, J. Degroote, B. Trachet, B. Verhegghe, P. Segers, J. Vierendeels, A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels, J. Comput. Appl. Math. 246 (2013) 10–17, http://dx.doi.org/10.1016/j.cam.2012.10.034, Fifth International Conference on Advanced Computational Methods in Engineering (ACOMEN 2011).

[79] X. Wang, T.S. Eriksson, T. Ricken, D.M. Pierce, On incorporating osmotic prestretch/prestress in image-driven finite element simulations of cartilage, J. Mech. Behav. Biomed. Mater. 86 (2018) 409–422, http://dx.doi.org/10.1016/j.jmbbm.2018.06.014.

[80] D. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), in: Y. Bengio, Y. LeCun (Eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016, http://dx.doi.org/10.48550/ARXIV.1511.07289.