

FREE UNIVERSITY OF BOZEN-BOLZANO  
FACULTY OF ENGINEERING

# **Hybrid Machine Learning and Numerical Analysis of Cartilage Biomechanics**

A dissertation submitted in partial fulfillment  
of the requirements for the degree of

Doctor of Philosophy in Computer Science

by

**Seyed Shayan Sajjadinia**

Supervised by Prof. Bruno Carpentieri  
Co-supervised by Prof. Gerhard A. Holzapfel

Under the evaluation of  
Prof. Andrea Manzoni (committee member)  
Prof. Bruno Carpentieri (committee member)  
Prof. Emiliiano Votta (reviewer and committee member)  
Prof. Nele Famaey (reviewer)

June 2024



... dedicated to my beloved parents.

# Abstract

Articular cartilage (AC), a crucial soft tissue for pain-free movement, undergoes significant biomechanical stress, warranting research through non-invasive computer simulations. Such simulations commonly use time-intensive numerical methods, such as finite element (FE) analysis. Artificial intelligence (AI), with machine learning (ML) as a surrogate model, presents a faster alternative, imitating numerical analysis with numerical data samples. Despite its potential, this method often requires extensive training and large datasets, which is inefficient. This study first introduces an efficient biomechanical model incorporating multi-physics modeling and a novel pre-stressing algorithm (PSA), generating few training samples across different fidelities and scales. We then propose efficient, multi-fidelity, generalizable surrogate models, enhanced by innovative preprocessing and training algorithms. Finally, the research code, including our developed Fortran subroutines and Python scripts, is open-sourced. Empirical results emphasize the significance of pre-stressing in multiphasic modeling and the value of efficient surrogate modeling in reducing computational time. The scalability and generalizability of the hybrid ML (HML) framework are confirmed through multiscale simulations, demonstrating the impact of our physics-constrained data augmentation (DA) and graph neural network (GNN) implementation. Therefore, while this study potentially advances the field with the developed PSA implementation, it also presents a straightforward approach to address computational challenges and data scarcity in ML-based surrogate modeling of cartilage biomechanics. Such research, though tailored to specific cartilage modeling, potentially paves the way for broader applications of our synergistic methodology through its open-source availability.

**Keywords:** Articular cartilage, knee biomechanics, finite element analysis, surrogate modeling, machine learning.

# Acknowledgements

I would like to express my deepest appreciation to my supervisors, Prof. Bruno Carpentieri and Prof. Gerhard A. Holzapfel, for their invaluable guidance and mentorship during my research journey. Their expertise and encouragement were not only pivotal in sparking my initial ideas but also played a crucial role in the growth and refinement of my work, greatly influencing the direction and depth of my papers. I am immensely grateful for their consistent support and joint authorship of the publications associated with this study, which has undoubtedly elevated the quality of my academic work.

Furthermore, my heartfelt thanks go to Prof. Shriram Duraisamy for his valuable contributions to the numerical aspects of my research and his co-authorship on two of the related publications. Lastly, I extend my sincere gratitude to the Free University of Bozen-Bolzano for their generous financial support and for providing the vital resources and educational programs that have been instrumental in advancing my project.

# Research Outputs

This work is based on the following relevant research outputs:

- **Journal papers:**

- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2021). “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”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 114, p. 104203. issn: 1751-6161.
- Sajjadinia, S.S., B. Carpentieri, D. Shriram, and G.A. Holzapfel (2022). “Multi-fidelity surrogate modeling through hybrid machine learning for biomechanical and finite element analysis of soft tissues”. In: *Computers in Biology and Medicine* 148, p. 105699. issn: 0010-4825.
- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (under review). “Bridging diverse physics and scales of knee cartilage with efficient and augmented graph learning”.

- **Conference proceedings papers:**

- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2021). “A pointwise evaluation metric to visualize errors in machine learning surrogate models”. In: *Proceedings of CECNet 2021*. Ed. by A.J. Tallón-Ballesteros. Vol. 345, *Frontiers in Artificial Intelligence and Applications*. IOS Press, pp. 26–34.
- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2024). “Large-scale finite element modeling of pre-stress in articular cartilage”. In: *Computer Methods in Biomechanics and Biomedical Engineering II. CMBBE 2023*. Ed. by W., Skalli, S., Laporte, A., Benoit. Vol. 39, *Lecture Notes in Computational Vision and Biomechanics*. Springer Nature Switzerland, pp. 105–112.

- **Invited book chapter:**

- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2024). “Hybrid data-driven and numerical modeling of articular cartilage”. In: *Big Data Analysis*

*and Artificial Intelligence for Medical Sciences.* Ed. by B. Carpentieri and P. Lecca. John Wiley & Sons, pp. 181–203.

- **Talk abstracts:**

- Sajjadinia, S.S., B. Carpentieri, D. Shriram, and G.A. Holzapfel (2021). “Biomechanical modeling of soft tissue multiphysics using hybrid machine learning and finite element analysis”. *17th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering*. Online.
- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2023). “Bridging tissue-scale multi-physics to organ-scale biomechanics through multi-fidelity machine learning”. *18th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering*. Paris, France.

- **Poster abstracts:**

- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2021). “A pointwise evaluation metric to visualize errors in machine learning surrogate models”. *The 3rd International Conference on Machine Learning and Intelligent Systems*. Online.
- Sajjadinia, S.S., B. Carpentieri, and G.A. Holzapfel (2023). “Large-scale finite element modeling of pre-stress in articular cartilage”. *18th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering*. Paris, France.

- **Research data:**

- Sajjadinia, S.S. (2021). “PMSE: Pointwise mean squared error”. Data corresponding to Chapter 2, available at [github.com/shayansss/pmse](https://github.com/shayansss/pmse).
- Sajjadinia, S.S. (2022). “HML: Hybrid machine learning”. Data corresponding to Chapter 4, available at [github.com/shayansss/hml](https://github.com/shayansss/hml).
- Sajjadinia, S.S. (2023). “PSA: Pre-stress algorithm”. Data corresponding to Chapter 3, available at [github.com/shayansss/psa](https://github.com/shayansss/psa).
- Sajjadinia, S.S. (2024). “EHML: Extended hybrid machine learning”. Data corresponding to Chapter 5, available at [github.com/shayansss/ehml](https://github.com/shayansss/ehml).

# Table of Contents

<b>Abstract</b>	<b>II</b>
<b>Acknowledgements</b>	<b>III</b>
<b>Research Outputs</b>	<b>IV</b>
<b>List of Acronyms</b>	<b>VIII</b>
<b>List of Figures</b>	<b>XIV</b>
<b>List of Tables</b>	<b>XVI</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Review of Preliminaries</b>	<b>4</b>
2.1 Cartilage Biomechanics . . . . .	4
2.1.1 Joint-Scale Properties . . . . .	4
2.1.2 Tissue-Scale Properties . . . . .	5
2.2 Physics-Based Modeling . . . . .	6
2.2.1 Governing Equations . . . . .	7
2.2.2 Implicit Finite Element Analysis . . . . .	11
2.3 Data-Driven Modeling . . . . .	14
2.3.1 Deep Learning . . . . .	14
2.3.2 Surrogate Modeling . . . . .	19
<b>3 Finite Element Analysis with Pre-stressing Optimizers</b>	<b>22</b>
3.1 Related Work . . . . .	22
3.2 Methodology . . . . .	24
3.2.1 Pre-Stressing Optimizers . . . . .	24
3.2.2 Simulation Tests . . . . .	27
3.3 Results and Discussion . . . . .	32

<b>4 Efficient Surrogate Modeling with Hybrid Multi-Fidelity Learning</b>	<b>39</b>
4.1 Related Work . . . . .	39
4.2 Methodology . . . . .	41
4.2.1 Efficient Hybrid Learning . . . . .	41
4.2.2 Simulation Tests . . . . .	42
4.3 Results and Discussion . . . . .	45
<b>5 Efficient and Generalizable Learning for Surrogate Modeling</b>	<b>52</b>
5.1 Related Work . . . . .	52
5.2 Methodology . . . . .	54
5.2.1 Efficient and Generalizable Learning . . . . .	54
5.2.2 Simulation Tests . . . . .	59
5.3 Results and Discussion . . . . .	63
<b>6 Conclusion</b>	<b>72</b>
<b>References</b>	<b>75</b>
<b>Appendix A Runtime Details</b>	<b>99</b>
<b>Appendix B Supplemental Results</b>	<b>106</b>

# List of Acronyms

<b>2D</b>	Two-Dimensional
<b>3D</b>	Three-Dimensional
<b>AC</b>	Articular Cartilage
<b>AE</b>	Autoencoder
<b>AI</b>	Artificial Intelligence
<b>CPU</b>	Central Processing Unit
<b>CR</b>	Compression Ratio
<b>DA</b>	Data Augmentation
<b>DL</b>	Deep Learning
<b>DZ</b>	Deep Zone
<b>FE</b>	Finite Element
<b>FFNN</b>	Feed-Forward Neural Network
<b>FPBBS</b>	Fixed-Point-Based Backward Scheme
<b>GNN</b>	Graph Neural Network
<b>GPU</b>	Graphics Processing Unit
<b>HF</b>	High-Fidelity
<b>HML</b>	Hybrid Machine Learning
<b>LF</b>	Low-Fidelity
<b>ML</b>	Machine Learning
<b>MO</b>	Material Optimization

<b>MOR</b>	Model Order Reduction
<b>MSE</b>	Mean Squared Error
<b>MZ</b>	Middle Zone
<b>PMSE</b>	Pointwise Mean Squared Error
<b>PSA</b>	Pre-stressing Algorithm
<b>ROM</b>	Reduced-Order Modeling
<b>SZ</b>	Superficial Zone

# List of Figures

2.1	Main substructures of a tibiofemoral model shown in different colors. . . . .	5
2.2	In the <i>in vivo</i> state of a circular cartilage plug, the angle between the radial axis and one of the primary fibrils is denoted as $\theta$ for illustration, based on Benninghoff 1925 and Wilson et al. 2004 studies. . . . .	6
2.3	Examples of finite elements and mesh entities: typically, the geometric domain is discretized into finite elements, and the boundary conditions are applied to the nodes. In this manner, integrals are computed at integration points and are then usually extrapolated to the nodes. The choice of element definition depends on dimensionality and physical characteristics, such as geometric complexity. . . . .	13
2.4	Example of a surrogate model and its evaluation. The normalized vertical Cauchy stress is assigned as the target values (as shown in the left contour subplot) and therefore is used to calculate errors. The evaluation of training performance is depicted using global metrics, with an emphasis on the widely recognized mean-squared error (MSE) metric. The $SE_m$ box plot highlights sample anomalies as well. The point-wise MSE (PMSE) contours are superimposed on the initial configuration. Observing the final snapshots of PMSEs indicates the inaccuracy in learning where the element shapes varied. PMSE offers valuable insights for further interpretability, yet MSE remains the conventional standard for metric comparison. For contour plots, the values are shown by a color range from blue to red, corresponding to -2750 to 165 MPa for stresses and 0 to 2 for PMSEs, respectively. . . . .	21

3.1	Role of the pre-stressing algorithm (PSA) in various simulation states: initially, the problem is presumed to start in a stress-free state with unknown state variables, encompassing geometrical configuration, material parameters, or both. Following the application of pre-stress, these state variables are updated to their revised values, aligning with the initial conditions of standard <i>in vivo</i> or <i>in vitro</i> tests that typically begin in a pre-stressed state. To ascertain the stress-free state, a PSA is employed, iteratively refining the initial estimates of state variables to align with the initial <i>in vivo</i> values. Once determined, the primary numerical analysis can proceed to simulate the experimental test. . . . .	25
3.2	Schematic of the pre-stressing algorithm, where the material state is illustrated by a blue shade, and blue and red arrows show the forward and backward analyses, respectively. At each step, pre-stressing is applied to achieve a new configuration, which is then compared with the reference state to calculate the new geometrical configuration, while the material state is also updated afterward. It is possible to optimize the material and geometrical states with separate optimizers or using a unified optimizer. . . . .	28
3.3	Schematic representation of the pre-stressing implementation in cartilage models, using the Fortran subroutines of Abaqus. . . . .	29
3.4	Flowchart of the backward optimization of our pre-stressing algorithm, executed in Abaqus using a Python script. . . . .	30
3.5	2D models with an axisymmetric geometry (left) and a plane strain geometry (right). . . . .	30
3.6	Example of the orientations of split lines of a cartilage substructure (left) and a local plane corresponding to a cartilage point (right). . . . .	31
3.7	Recorded stress vs. stretch extracted from calibrated numerical models mimicking an experimental test [Elliott et al. 2002]. Abbreviations: MO = material optimization; FE: finite element. . . . .	33
3.8	Contour of a representative fibrillar angle $\theta$ with respect to the radial axis in the pre-stressed state. Abbreviation: MO = material optimization. . . . .	34
3.9	Comparative plots of volume fractions with respect to the cartilage normalized depth in the pre-stressed initial state compared with the experimental values [Wilson et al. 2007; Rieppo et al. 2004; Shapiro et al. 2001; Lipshitz et al. 1975]. Abbreviations: MO = material optimization; FE = finite element. . . . .	35
3.10	Stress vs time plots from numerical simulations of unconfined compression experiments with (black curves) and without (red curves) material optimization for different cartilage zones and effective solid parts after equilibrium. . . . .	36

3.11 Total stress vs strain plots from numerical simulations of unconfined compression experiments with (black curves) and without (red curves) material optimization across different cartilage zones. . . . .	36
3.12 Contour plot of pre-stressed cartilage displacements. Note that the displacement values on the cartilage surface accumulate due to the deformation in the underlying layers, resulting in large absolute displacement values. However, this does not necessarily indicate correspondingly large local deformations. These relatively large values are comparable to the regular displacements in the knee cartilage tissue [Chan et al. 2009]. . . . .	37
3.13 Contour plots of stresses in the 3D pre-stressed state. Since the superficial-zone values are negligible, they are not shown here. It is important to note that the constitutive equation for the fibrils is based on the highly discrete orientation of the fibrillar bundles (eq. 2.23), leading to non-physiological concentrations of stress in the model. To mitigate this issue, the selection of higher-order elements or the adoption of a denser mesh could have been beneficial. However, given the good agreement of the results with the previous findings, the use of more computationally expensive meshes was deemed unnecessary. . . . .	38
4.1 Illustration of a standard machine learning surrogate model (a) and its integration in a hybrid machine learning (HML) framework, as connected with a simplified finite element model sequentially (b). . . . .	41
4.2 Illustration of various small-scale computational samples for surrogate modeling. . . . .	42
4.3 Numerical mesh of the large-scale problem used for creating surrogate models.	43
4.4 Representation of the relative decrease in runtime for low-fidelity simulations compared to high-fidelity simulations. This data is obtained by calculating the ratio of the runtime reductions to the runtimes of the low-fidelity simulations. . . . .	46
4.5 Assessment of trained models with different numbers of training samples in 2D tests, implemented via machine learning (ML) and hybrid ML (HML).	46
4.6 Comparative analysis of loss values against the number of epochs in 3D surrogate modeling. It is observed that certain models achieved convergence earlier owing to the existence of the early stopping mechanism. Abbreviations: ML = machine learning; HML = hybrid machine learning. . . . .	48

4.7 Illustration of pointwise mean squared error (PMSE) contours for all three-dimensional surrogate models. Compared to regular machine learning (ML), the hybrid ML (HML) achieves satisfactory accuracy in most areas; however, high local errors may persist at a few nodes. These errors are likely due to numerical modeling inaccuracies or the compromises made during manual hyperparameter tuning. Specifically, hyperparameters were chosen based on their ability to rapidly converge and for their capability to reduce the order of the latent space efficiently, allowing models to complete training within a few minutes. While this approach helps maintain focus on general model performance, it might overlook some local complexities. With higher-quality data and a more intensive hyperparameter search, these local errors could potentially be minimized. . . . .	50
5.1 Example of a central node of a one-hop subgraph, shown in red, interacting with neighboring nodes, shown in blue, within a numerical cartilage model. By subgraphing, the semantics can be preserved in the randomly selected central (target) nodes. . . . .	56
5.2 Sample of data augmentation in the spatial and temporal domains, with the points constrained to move on possible or allowable directions. . . . .	57
5.3 Sample of a model with and without data augmentation. . . . .	58
5.4 Key stages of the custom training loop. The primary distinction from standard training loops is the incremental preprocessing of large batches of graphs at each epoch, coupled with iterative training within a separate inner loop for these large batches; thereby speeding up the training process. Additionally, the early stopping algorithms and learning rate schedulers are employed to prevent overshooting and overfitting, respectively. . . . .	58
5.5 Illustration of the role of different models at different scales in surrogate modeling. . . . .	59
5.6 Comparison of data generation and preprocessing pipelines for hybrid machine learning (a) and its extended version (b). For both methods, numerical sampling is executed by independent uniform distributions (considering that the input spaces are not extremely large). Furthermore, the extended version does not disregard the inner temporal frames, thus requiring synchronization through temporal interpolation. It also augments the dataset with geometric noises after data generation, allowing for training samples to be gathered from simple and structured meshes. . . . .	61

5.7 Comparison of various preprocessing transformations on nodal input and output features. These transformations were applied to each feature dimension independently but were aggregated here solely for illustration purposes. Overall, the transformations alone could provide a more standardized normalized distribution, while their combination yielded a more uniform distribution. This approach can make the average evaluation errors highly sensitive to a small minority of complex nodes, rather than the majority of simpler nodes, which are typically in areas far from the boundaries (having small values). The existence of many small but few large values is attributable to the nature of soft tissues, where only small regions deform and resist pressure significantly more than other areas, particularly when the deformation is moderate but applied rapidly, as implemented in this study. . . . .	65
5.8 Comparative analysis of validation errors across training epochs for the models without the autoencoder. Abbreviation: CR = compression ratio. . . . .	67
5.9 Comparative analysis of validation errors across training epochs for the models with the autoencoder. Abbreviation: CR = compression ratio. . . . .	68
5.10 Averaged out-of-distribution testing errors for various models trained using different loss functions. Abbreviation: AE = autoencoder. . . . .	69
5.11 Averaged (out-of-distribution testing) pointwise mean-squared errors (PM-SEs) for various models trained using different loss functions. . . . .	70

# List of Tables

3.1	Fitted constitutive parameters of the fibrillar network for different implementations. Abbreviations: MO = material optimization; PSA = pre-stressing algorithm. . . . .	32
4.1	Recorded mean-squared errors ( $\times 10^{-3}$ ) on validation/testing sets in 2D modeling, implemented via machine learning (ML) and hybrid ML (HML). . . . .	47
4.2	Loss values at the final epoch on training and validation sets in 3D modeling, implemented via machine learning (ML) and hybrid ML (HML). . . . .	49
5.1	Runtimes (Min) with their mean and standard deviation for each model. . . . .	63
A.1	Runtime details on a regular CPU for generating small-scale numerical samples used in Chapter 4. . . . .	99
A.2	Runtime details on a regular CPU for generating large-scale numerical samples used in Chapter 4. . . . .	100
A.3	Runtime details on a regular CPU for generating small-scale numerical samples used in Chapter 5. . . . .	103
A.4	Runtime details on a regular CPU for generating large-scale numerical samples used in Chapter 5. . . . .	105
B.1	In-distribution mean-squared error of the small-scale model used in Chapter 5 with message passings: 2. Abbreviations: AE = autoencoder; CR = compression ratio. . . . .	106
B.2	In-distribution mean-squared error of the small-scale model used in Chapter 5 with message passings: 5. Abbreviations: AE = autoencoder; CR = compression ratio. . . . .	106
B.3	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 4 and number of message passings 2. Abbreviation: AE = autoencoder. . . . .	107
B.4	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 4 and number of message passings 5. Abbreviation: AE = autoencoder. . . . .	107

B.5	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 2 and number of message passings 2. Abbreviation: AE = autoencoder. . . . .	107
B.6	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 2 and number of message passings 5. Abbreviation: AE = autoencoder. . . . .	108
B.7	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 1 and number of message passings 2. Abbreviation: AE = autoencoder. . . . .	108
B.8	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 1 and number of message passings 5. Abbreviation: AE = autoencoder. . . . .	108
B.9	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 0.5 and number of message passings 2. Abbrevia- tion: AE = autoencoder. . . . .	109
B.10	Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 0.5 and number of message passings 5. Abbrevia- tion: AE = autoencoder. . . . .	109

# Chapter 1

## Introduction

Articular cartilage (AC), a soft tissue that provides smooth load-bearing between articulating bones with excellent lubrication, allows for pain-free movement. This tissue, under normal conditions, shows remarkable shock-absorbing properties and resistance [X.L. Lu et al. 2008]. However, biomechanical factors, such as extreme loads on the tissue, can cause significant damage to it, leading to substantial health care expenditures [Salmon et al. 2016]. Understanding the biomechanics of cartilage is therefore crucial, particularly for early damage detection or prevention [Martínez-Moreno et al. 2019].

In this regard, artificial intelligence (AI), or more precisely machine learning (ML), has been widely adopted. Recent research has demonstrated progress in various aspects: detection of damage via biomechanical markers [Alunni Cardinali et al. 2023], location-dependent material characterization of cartilage [Hamsayeh Abbasi Niasar et al. 2023], and prediction of the tissue's fibrillar orientation [Mirmojarabian et al. 2023]. Nonetheless, gathering clinical or experimental data for generating (training) these ML models remains a significant hurdle, leading to a preference for physics models (which, due to their complexity, are commonly approximated by numerical approaches).

Finite element (FE) modeling, a prominent type of numerical modeling in this context, breaks down the complex physics domain into smaller, manageable elements, used for numerical approximation of the solution. This method is applied in diverse scenarios, such as simulating the fibrillar orientation [Sajjadi et al. 2021], and analyzing the hyperphysiological (potentially damaging) compression [Occhetta et al. 2019], and image-based modeling in human joints [Thienkarochanakul et al. 2020]. Despite their effectiveness, these methods involve iterative algorithms that can be computationally intensive, with processing times ranging from minutes to even days [Haut Donahue et al. 2002; Kazemi et al. 2011; Naghibi Beidokhti et al. 2016; M. Wang et al. 2018; Lostado Lorza et al. 2021].

To mitigate the high computational demands of the numerical methods, ML has been commonly employed to develop data-driven replacements of numerical models, known as surrogates. These surrogate models, which are trained on the numerically generated data samples, can significantly reduce computation time, as demonstrated in multitudes of

cartilage studies [Paiva et al. 2012; Arbabi et al. 2016a; Arbabi et al. 2016b; Egli et al. 2021]. Despite these advantages, they still face significant limitations, including a lack of generalizability and the need for a large number of training samples (which leads to an oversimplification of the numerical model to efficiently generate them). Addressing these issues is the main motivation for this work.

The aim of this project is to develop a new computer simulation tool that balances computational cost and the fidelity of a numerical model across various scales. To achieve this, the complex mathematical models will be optimized for efficient and advanced implementation of key cartilage physics while maintaining sufficient accuracy. Subsequently, highly efficient machine learning ML surrogates will be developed to expedite simulations, overcoming the aforementioned limitation. Accordingly, the research questions posed are as follows:

1. How can a complex multi-physics model of cartilage biomechanics be implemented efficiently enough to generate the required training datasets?
2. How can machine learning techniques be integrated into the numerical model for efficient surrogate modeling?
3. Is the new machine learning system scalable and generalizable across different scales?
4. What are the potential implications of the proposed methods for future research?

In the subsequent chapter, we provide a comprehensive overview of the theoretical foundations, encompassing aspects of numerical analysis, biomechanics, and data-driven methodologies, culminating with a simplified illustration of a surrogate model to aid in understanding. Following this, the chapter thereafter we regard the first research question, by presenting the development of a novel complex yet fully automated multi-physics model. It is particularly emphasized that special consideration must be given to the tissue pre-stressing, which will be demonstrated to be crucial for high-fidelity (HF) simulations across various scales.

In light of the third research question, Chapter 4 introduces a novel hybrid ML (HML), integrating ML with a reduced-order, low-fidelity (LF), numerical approach. This model, as we will demonstrate, achieves sufficient accuracy even with limited HF data and boasts efficient training processes, thereby addressing the third research question. Next, regarding Chapter 5, we further enhance this HML technique by incorporating additional techniques such as a new data augmentation (DA), improving its ability to generalize across different scales and thus responding to the third question. In other words, this chapter further enhances the single-scale few-shot learning and evaluates the model's potential in achieving zero-shot generalizability.

The final chapter summarizes the innovative contributions of this work and provides direct responses to the research questions, including the last question regarding the possible future directions.

# Chapter 2

## Review of Preliminaries

This work draws from diverse disciplines, each of which may be the primary focus of distinct research groups. Recognizing that many readers might not be thoroughly versed in all these areas, this chapter aims to present an overview of the foundational concepts, with the assumption that readers have a basic understanding of tensor calculus [Fleisch 2011], linear algebra [Strang 2023], and stochastic programming [Birge et al. 2011]. The opening section elucidates the principles of cartilage biomechanics, exploring both joint scale and tissue scale properties. Subsequent sections delve into the complexities of implicit FE analysis and its mathematical intricacies that are used to generate the datasets. The final section unfolds machine learning techniques, highlighting their prominence in this context. This chapter is finished, by implementing a simplified and educational example.

### 2.1 Cartilage Biomechanics

#### 2.1.1 Joint-Scale Properties

The human knee is a complex structure, considering that it houses the tibiofemoral joint, the largest human joint. This joint is a union of the femur and tibia bones, intertwined with cartilage, ligaments, and menisci (Fig. 2.1). These are the soft tissues that safeguard the joint from abnormal movements and potential harm, stabilizing the joint movement and reducing the concentrated pressure [Walker et al. 1975; Mameri et al. 2022]. This unique cam-shape design, perfected by evolution, accommodates the asymmetrical and complex musculoskeletal movements executed by the human body [Dye 1987; Goldblatt et al. 2003]. Nonetheless, damage to each of these tissues might induce cartilage degeneration, known as osteoarthritis.

Osteoarthritis is a degenerative ailment that affects cartilage, causing pain, discomfort, and eventual joint malfunction [Lespasio et al. 2017]. Predominantly seen in older individuals, especially women, this condition exerts considerable strain on health organizations due to its socioeconomic implications [A. Chen et al. 2012; Hawker 2019]. Multiple fac-

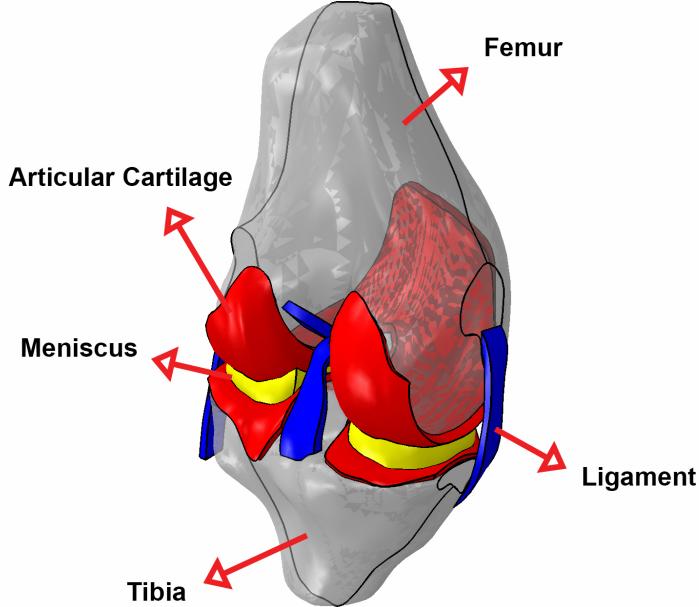


Figure 2.1: Main substructures of a tibiofemoral model shown in different colors.

tors, ranging from genetics to lifestyle choices, contribute to its onset [Loeser et al. 2016; Mobasheri et al. 2017; Astephen Wilson et al. 2021]. To analyze this disease, it is essential to first understand cartilage biomechanics regardless of its damage. Consequently, a significant portion of the research, including this study, has focused on this aspect [Kong et al. 2022].

Similarly to other connective tissues, the functional attributes of AC predominantly stem from its extracellular elements, particularly proteoglycans and collagen network [Culav et al. 1999; Brody 2015]. Given this, biomechanical investigations heavily concentrate on modeling how these elements interplay [Klika et al. 2016; Ebrahimi et al. 2019; Sajjadinia et al. 2019; Lin et al. 2021; Paz et al. 2022]. This topic is further explored in the following subsection.

### 2.1.2 Tissue-Scale Properties

AC is an intricate mix of solid and fluid constituents. Around 60 – 80% of the tissue is water, brought about by the cartilage’s osmotic pressure (caused by proteoglycans) and its porous architecture [Cederlund et al. 2022]. This water content does not just ensure smooth lubrication on the cartilage surface but also presents formidable resistance against applied loads on tissue, more than most other microstructural components [Quiroga et al. 2017; Sajjadinia et al. 2019]. Therefore, it is common to assume a biphasic composition forms AC with a fluid phase (water) and an effective solid phase (solid components with other biomechanical parameters like osmotic pressure).

Aggrecans, the most prevalent cartilage proteoglycans, are attached to hyaluronic acid

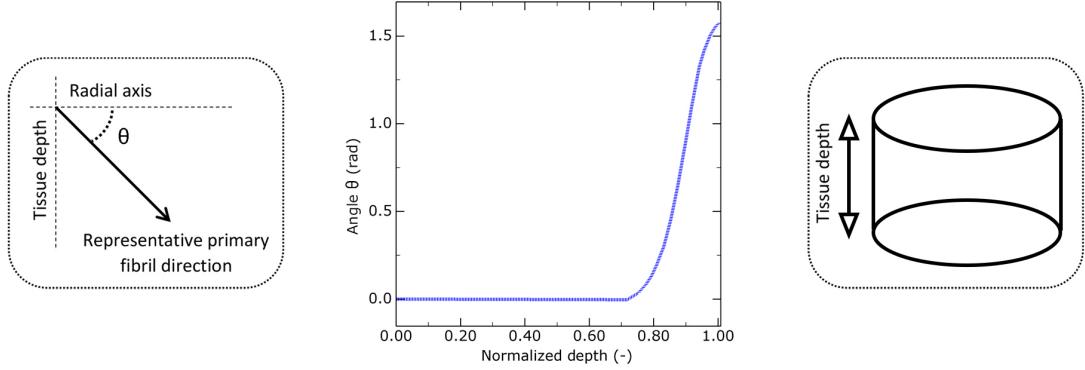


Figure 2.2: In the *in vivo* state of a circular cartilage plug, the angle between the radial axis and one of the primary fibrils is denoted as  $\theta$  for illustration, based on Benninghoff 1925 and Wilson et al. 2004 studies.

chains, creating aggregates within the collagen network. Owing to its glycosaminoglycan components, these aggregates carry fixed negative charges. This establishes a chemical gradient, causing water attraction through the osmosis mechanism, explaining the presence of water in the tissue [Kiani et al. 2002; Gómez-Florit et al. 2020; Johnson et al. 2021]. This results in the above-mentioned osmotic pressure which, by counteracting external forces, plays a role in the tissue’s load-bearing ability. Furthermore, this internal pressure increases tissue size and stretches the collagen fibrils, resulting in reversible tissue deformation and higher stiffness [Dudhia 2005].

The extracellular matrix is reinforced by collagen fibrils, which protect AC against tensile strains. This is analogous to the reinforcement provided by structural cables, marking it as another significant solid component to consider [Bozec et al. 2007; Bielajew et al. 2020]. These fibrils can be categorized into primary (anisotropic) and secondary (isotropic) groups [Clark 1985; Wilson et al. 2004]. The latter, in healthy tissue, run in all spatial directions, while the former have an arcade-like orientation (Fig. 2.2), starting vertically from the deeper AC sections and undergoing a mid-course twist to run parallel to the cartilage surface [Wilson et al. 2004]. These superficial fibrils help protect the surface from shear stresses [Shirazi et al. 2008; Motavalli et al. 2014]. Therefore, these fibrillar bundles are commonly integrated into mathematical models of cartilage, along with the aforementioned components, as will be discussed in the next section.

## 2.2 Physics-Based Modeling

Note that in the following of this work, we may very occasionally use the same mathematical symbols for different purposes. This is an intentional choice, made to avoid the use of non-standard symbols, and the context can clarify their meanings. Also, note that while “ $:=$ ” signifies a definition, “ $:$ ” represents the double dot product.

### 2.2.1 Governing Equations

In physics-based modeling of AC, inertia and weight are often overlooked due to the tissue's low-density structure [Pearle et al. 2005]. Newton's second law suggests that any force applied to the tissue, or surface traction, will reach equilibrium with other applied forces, leading to cartilage deformation. Below is a simplified representation of the 3D equilibrium equation for this phenomenon:

$$\int_s \mathbf{T} dS = \mathbf{0}, \quad (2.1)$$

where the surface area of the deformed solid material is  $S$ , and the associated traction on this surface,  $\mathbf{T}$ , is related to the full Cauchy stress tensor  $\boldsymbol{\sigma}_T$  as:

$$\mathbf{T} = \boldsymbol{\sigma}_T \cdot \mathbf{n}, \quad (2.2)$$

with  $\mathbf{n}$  being the external normal to the surface. The stress tensor provides a general view of how forces are distributed at each location. Assuming stress as a field parameter and according to the divergence theorem, the flux of the field on a closed surface can be correlated to the field's spatial divergence within the volume  $V$ . Thus:

$$\int_S \boldsymbol{\sigma}_T \cdot \mathbf{n} dS = \int_V \nabla_{\mathbf{x}} \boldsymbol{\sigma}_T dV, \quad (2.3)$$

where  $\nabla_{\mathbf{x}}$  denotes the gradient operator with respect to the position vector after deformation, i.e.,  $\mathbf{x}$ . From this, the following differential equation can be deduced without integration [Holzapfel 2000]:

$$\nabla_{\mathbf{x}} \boldsymbol{\sigma}_T = \mathbf{0}. \quad (2.4)$$

For considering the biomechanical role of the fluid phase, the continuity equation can be used. It implies that the rate of change of fluid within the tissue is equal to the amount passing the boundary, based on the premise that AC is a water-saturated porous medium:

$$\frac{d}{dt} \left( \int_V \rho \phi^F dV \right) + \int_S \rho \phi^F \mathbf{n} \cdot \mathbf{v}_r dS = 0, \quad (2.5)$$

where  $\mathbf{v}_r$  is the relative fluid velocity with respect to the solid framework,  $\rho$  represents fluid mass density, and  $\phi^F$  is the local fluid volume fraction. By inserting the volume ratio  $J$  in the first term (to account for the solid phase deformation) and applying the divergence theorem to the second term, we get the following partial differential equation:

$$\frac{1}{J} \frac{d}{dt} (J \rho \phi^F) + \nabla \cdot (\rho \phi^F \mathbf{v}_r) = 0. \quad (2.6)$$

In the next step, the material (constitutive) equations will be defined using finite strain

theory, as explained in the following reference [Holzapfel 2000].

We assume that  $\nabla_{\mathbf{X}}$  is the gradient operator with respect to the position vector before deformation,  $\mathbf{X}$ . If  $\mathbf{I}$  represents the identity tensor, then the deformation gradient  $\mathbf{F}$  is:

$$\mathbf{F} = \nabla_{\mathbf{X}} \mathbf{u} + \mathbf{I}. \quad (2.7)$$

This second-order tensor is a metric of relative deformation, which can be split into the rotation tensor  $\mathbf{R}$  and the symmetric tensor that is left stretch tensor  $\mathbf{V}$ :

$$\mathbf{F} = \mathbf{VR}. \quad (2.8)$$

To eliminate the effects of rigid body rotation on tissue stress and deformation (since inertia effects are ignored), we use the orthogonality of the rotation tensor, i.e.,  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ . Accordingly, the left Cauchy-Green deformation tensor  $\mathbf{B}$  is defined to encompass only the stretches, as follows:

$$\mathbf{B} = \mathbf{FF}^T = \mathbf{V}^2. \quad (2.9)$$

In this way, the volume ratio can also be determined as:

$$J = \sqrt{\det \mathbf{B}}. \quad (2.10)$$

Next, the logarithmic strain, denoted as  $\boldsymbol{\epsilon}$ , serves as another measure of deformation and is defined by:

$$\boldsymbol{\epsilon} := \ln \mathbf{B}. \quad (2.11)$$

Subsequently, we can use Hook's law, the simplest (linear and isotropic) constitutive model, as follows:

$$\boldsymbol{\sigma} := \frac{1+\nu}{E} \left[ \boldsymbol{\epsilon} + \frac{\nu}{1-2\nu} (\text{tr}(\boldsymbol{\epsilon}) \mathbf{I}) \right]. \quad (2.12)$$

Here, Young's modulus  $E$  and Poisson's ratio  $\nu$  are the material constants. A superior alternative is the nearly incompressible neo-Hookean model, which exhibits some nonlinearity during extensive deformations akin to rubber-like materials:

$$\boldsymbol{\sigma} := \frac{2}{J} \mathbf{F} \frac{\partial \Psi}{\partial \mathbf{C}} \mathbf{F}^T, \quad \text{where } \Psi = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1}(J - 1)^2. \quad (2.13)$$

Here,  $\Psi$  represents the empirically defined strain-energy function. The right Cauchy-Green tensor is denoted by  $\mathbf{C}$  and is related by the equation  $\mathbf{C} = \mathbf{F}^T \mathbf{F}$ . Moreover, the first invariant of the deviatoric part of  $\mathbf{C}$  is represented as  $\bar{I}_1$ . Notably, this stress definition does not account for the different cartilage phases essential for achieving HF results [Holzapfel 2000].

To model the interaction between the solid and fluid phases, we employ the well-

recognized porous media theory in conjunction with Darcy's law. This law posits that the fluid transition from an area of elevated pressure to one of diminished pressure is determined by the constitutive behavior, such as the dynamic viscosity  $\mu$ , i.e., the fluid's inherent resistance to flow [Terzaghi 1943; Dullien 1979]:

$$\phi^F \mathbf{v}_r = -\frac{1}{\mu} \mathbf{K} \cdot \nabla_{\mathbf{x}} P, \quad (2.14)$$

where the fluid volume fraction is denoted by  $\phi^F$  and the permeability tensor  $\mathbf{K}$  determines the porous structure's ability to permit fluid flow within the solid matrix. This results in fluid pressure  $P$  and effective solid stress  $\boldsymbol{\sigma}^{\text{EFF}}$  as follows:

$$\boldsymbol{\sigma} := \boldsymbol{\sigma}^{\text{EFF}} - PI. \quad (2.15)$$

The constrained mixture theory [Klisch 1999] is frequently adopted to account for the contributions of different phases to the effective stress in AC. According to this theory, non-fluid segments are bound to each other, leading them to exhibit similar deformation. This is expressed as:

$$\boldsymbol{\sigma}^{\text{EFF}} := \boldsymbol{\sigma}^{\text{COL}} + \boldsymbol{\sigma}^{\text{MAT}} - \boldsymbol{\sigma}^{\text{GAG}}. \quad (2.16)$$

Within this framework, the annotations COL, MAT, and GAG denote the stress contributions from the collagen network, the non-fibrillar solid matrix, and the osmotic pressure, respectively [Mow et al. 1980; Wilson et al. 2005; Sajjadinia et al. 2019].

Regarding the osmotic pressure, it can be approximated by the electrostatic forces of glycosaminoglycans. This force is also strictly associated with solid matrix deformation [Ateshian et al. 2004]. Making this assumption simplifies the expression to an exponential form with material constants  $\alpha_1$  and  $\alpha_2$  [Ateshian et al. 2004; Buschmann et al. 1995; Stender et al. 2013]:

$$\boldsymbol{\sigma}^{\text{GAG}} = \alpha_1 J^{-\alpha_2} \mathbf{I}. \quad (2.17)$$

This equation is based on the Poisson-Boltzmann cell model, which assumes that the internal pressure of cartilage is primarily generated by chemical-expansion stress and is better described by a molecular-level description rather than by a macroscopic Donnan model [Hill 1956; Buschmann et al. 1995; Deserno et al. 2001].

As to the collagen fibrils, a nonlinear relationship between loads and strains is evident through experimentation [Charlebois et al. 2004]. Focusing on a specific fibrillar bundle  $I$ , this relationship can be represented as  $(E_1 + E_2 \epsilon^I) \epsilon^I$ , with  $E_1$  and  $E_2$  highlighting the nonlinearity extent. Then,

$$\epsilon^I = \log(\lambda^I). \quad (2.18)$$

Here, the fibrillar stretch is represented by  $\lambda^I$ , which is related to the total deformation

by

$$\lambda^I = \sqrt{\mathbf{n}^I \cdot \mathbf{B} \cdot \mathbf{n}^I}, \quad (2.19)$$

where,  $\mathbf{n}^I$  is the new direction of the fibril bundle  $I$  after deformation, which is correlated with the initial direction vector  $\mathbf{N}^I$  via

$$\mathbf{n}^I = \frac{\mathbf{F} \cdot \mathbf{N}^I}{\|\mathbf{F} \cdot \mathbf{N}^I\|_2}, \quad (2.20)$$

where  $\|\bullet\|_2$  denotes the  $l^2$  norm of  $\bullet$ . Furthermore, to account for surface area variations, the aforementioned constitutive equation of fibril  $I$  can be multiplied with  $\lambda^I/J$ , the inverse of the surface area of the fibril bundle, yielding

$$\sigma^I = \phi_0^S \rho_C^I \frac{\lambda^I}{J} (E_1 + E_2 \epsilon^I) \epsilon^I \quad \text{if } \epsilon^I > 0, \quad (2.21)$$

where,  $\rho_C^I$  represents the volume fraction of pertinent fibrils and  $\phi_0^S$  indicates the solid's initial volume fraction. The volume fractions are inserted in the equation to encompass the contribution of each phase based on mixture theory. Considering the compressibility of the porous solid structure of cartilage, these values are updated by the continuity equation, as follows

$$\varphi = \frac{\varphi_0}{J} \quad \varphi \in \{\phi^S, \rho^{\text{COL}}\}. \quad (2.22)$$

Here,  $\rho^{\text{COL}}$  is the total collagen volume fraction. Now, the total stress in the collagen fibrils at all directions, expressed as  $\boldsymbol{\sigma}^{\text{COL}}$  can be determined by [Wilson et al. 2004]:

$$\boldsymbol{\sigma}^{\text{COL}} = \sum_{I=1}^9 \sigma^I \mathbf{n}^I \otimes \mathbf{n}^I, \quad (2.23)$$

where  $\otimes$  represents the dyadic product.

Finally, the other parts of the solid material can be modeled by nonlinear isotropic elastic models. One such common model is the neo-Hookean equation, grounded in rubber-like material thermodynamics [Kim et al. 2012]. This equation is adjusted to reflect the volume fractions of the material components [Wilson et al. 2007; Sajjadinia et al. 2019]:

$$\boldsymbol{\sigma}^{\text{MAT}} = \phi_0^S G_m \frac{1 - \rho^{\text{COL}}}{J} \left[ \frac{\ln J}{2} \left( \frac{1}{3} + \frac{J + \phi_0^S}{J - \phi_0^S} - \frac{\phi_0^S J \ln J}{(J - \phi_0^S)^2} \right) \mathbf{I} + (\mathbf{B} - J^{2/3} \mathbf{I}) \right]. \quad (2.24)$$

Here,  $G_m$  is a material constant. Hence, the stress in the components of this HF multi-physics model correlates with the deformation in the tissue.

The multiphasic model can be further simplified either by ignoring the contributions of some of its components or by using an alternative monophasic (elastic) model. An additional possible assumption in monophasic models is to incorporate a level of viscosity to model the time-dependent response (without including the fluid phase that makes

biphasic models time-dependent). To exemplify a viscoelastic model, we first denote the deviatoric (DEV) and hydrostatic parts of a symmetrical second-order tensor  $\mathbf{A}$  with superscripts  $D$  and  $H$ , respectively. Accordingly, we have [Backus 1970]

$$\text{DEV}(\mathbf{A}) = \mathbf{A} - \mathbf{A}^H, \quad \text{where } \mathbf{A}^H = \frac{1}{3}\text{tr}(\mathbf{A})\mathbf{I}. \quad (2.25)$$

Defining the Kirchhoff stress  $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ , its instantaneous value (without viscosity) is denoted by the subscript 0. Then, a simplified viscoelastic model is formulated as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 - \text{DEV} \left[ \frac{\mathcal{G}}{\mathcal{T}} \int_0^t \bar{\mathbf{F}}_t^{-1}(t-s) \cdot \text{DEV}(\boldsymbol{\tau}_0(t-s)) \cdot \bar{\mathbf{F}}_t^{-T}(t-s) e^{-s/\mathcal{T}} ds \right] \quad (2.26)$$

$$- \frac{\mathcal{K}}{\mathcal{T}} \int_0^t \boldsymbol{\tau}_0^H(t-s) e^{-s/\mathcal{T}} ds. \quad (2.27)$$

Here,  $\bar{\mathbf{F}}_t^{-1}(t-s) = \bar{\mathbf{F}}(t) \cdot \bar{\mathbf{F}}^{-1}(t-s)$  and its transposed are used as a transformation operator to shift the stress tensor to the current configuration (as it can be rotated for the deviatoric part), where  $\bar{\mathbf{F}} = J^{-1/3}\mathbf{F}$  is the distortion gradient (disregarding the volume change since  $\det \bar{\mathbf{F}} = 1$ ). Also,  $\mathcal{G}$  and  $\mathcal{K}$  are material constants controlling the time-dependent shear and bulk moduli, respectively, while  $\mathcal{T}$  is another constitutive constant controlling the tissue relaxation time until equilibrium is reached [Fazekas et al. 2018; Abaqus 2021; Simo 1987].

### 2.2.2 Implicit Finite Element Analysis

The algorithms behind FE analysis typically make use of the weak formulation of differential equations. For this reason, the virtual work principle [Antman et al. 1979] can be applied to the eq. (2.4), yielding to

$$\int_V \boldsymbol{\sigma}_T \nabla_{\mathbf{x}} \cdot \delta \mathbf{u} dV = 0, \quad (2.28)$$

with  $\delta \mathbf{u}$  signifying the virtual displacement vector contributing to the virtual work. This way, the strong form, eq. 2.4, is transformed into a simpler equation with scalar output due to the existence of the integrals and dot product. Next, by applying integration by parts, the divergence theorem, and leveraging eq. 2.2, boundary, and volumetric parts are distinguished, as follows [Holzapfel 2000; Belytschko et al. 2014]:

$$\int_V \delta \mathbf{u} \nabla_{\mathbf{x}} \boldsymbol{\sigma}_T dV - \int_S \mathbf{T} \delta \mathbf{u} dS = 0. \quad (2.29)$$

In case the fluid phase is also modeled, we first discretize the time derivatives using finite differences, particularly through the backward Euler approach. For instance, applying

this to a function  $y$ , in this case the first term of eq. 2.6, results in

$$\frac{dy}{dt} \Big|_{t+\Delta t} \approx \frac{y_{t+\Delta t} - y_t}{\Delta t}. \quad (2.30)$$

After finding the derivative and then finding the weak form with a method similar to the weak form derivation of the equilibrium equation, a weak formulation for this equation is also obtained using

$$\begin{aligned} & \int_V \left( [\delta P]_{t+\Delta t} \left( [J\rho\phi^F]_{t+\Delta t} - [J\rho\phi^F]_t \right) - \Delta t [\rho\phi^F \nabla_x \delta P \cdot \mathbf{v}_r]_{t+\Delta t} \right) dV \\ & + \Delta t \int_S [\delta P \rho\phi^F \mathbf{n} \cdot \mathbf{v}_r]_{t+\Delta t} dS = 0, \end{aligned} \quad (2.31)$$

where  $\delta P$  indicates the virtual fluid pressure. Combining this with eq. 2.14 gives

$$\begin{aligned} & \int_V \left( [\delta P]_{t+\Delta t} \left( [J\rho\phi^F]_{t+\Delta t} - [J\rho\phi^F]_t \right) - \Delta t [\rho\phi^F \nabla_x \delta P \cdot \mathbf{v}_r]_{t+\Delta t} \right) dV \\ & - \Delta t \int_S \left[ \delta P \frac{\rho}{\mu} \mathbf{n} \cdot \mathbf{K} \cdot \nabla_x P \right]_{t+\Delta t} dS = 0. \end{aligned} \quad (2.32)$$

This equation, together with eq. 2.29 with one of the solid constitutive equations applied, defines the possible weak formulation in which the integrals can be calculated numerically after discretizing the physics domain into smaller elements, called FEs.

For clarification of the spatial discretization in FE analysis, a nonlinear 1D function,  $f(x)$ , is exemplified by its approximation after discretization,  $\bar{f}(x)$ :

$$\bar{f}(x) = \sum_{i=1}^n \mathcal{N}^i(x) f(x_i), \quad (2.33)$$

where the spatial space is divided into 1D FEs with nodal inputs  $x_i$  associated with outcomes  $f(x_i)$  over the domain  $[x_{i-1}, x_i]$  of element  $i$ . The defined shape functions of the element,  $\mathcal{N}^i(x)$ , are then expressed on two nodes, as follows:

$$\mathcal{N}_1^i(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}} \quad \text{and} \quad \mathcal{N}_2^i(x) = \frac{x_i - x}{x_i - x_{i-1}}. \quad (2.34)$$

These shape or interpolation functions represent  $f(x)$  as a sequence of interconnected simpler functions at the nodal points. In practice, the interpolation can be of higher order and dimensions, allowing more accurate numerical integrations, as illustrated in Fig. 2.3.

Extending the concept of 1D FEs to 3D space of this multi-physics equation, shape functions,  $\mathcal{N}^P$  and  $\mathcal{N}^u$ , are introduced similarly for spatial discretization of  $P$  and  $\mathbf{u}$  (along with their virtual variations). Inserting them into eqs. 2.32 and 2.29, we get

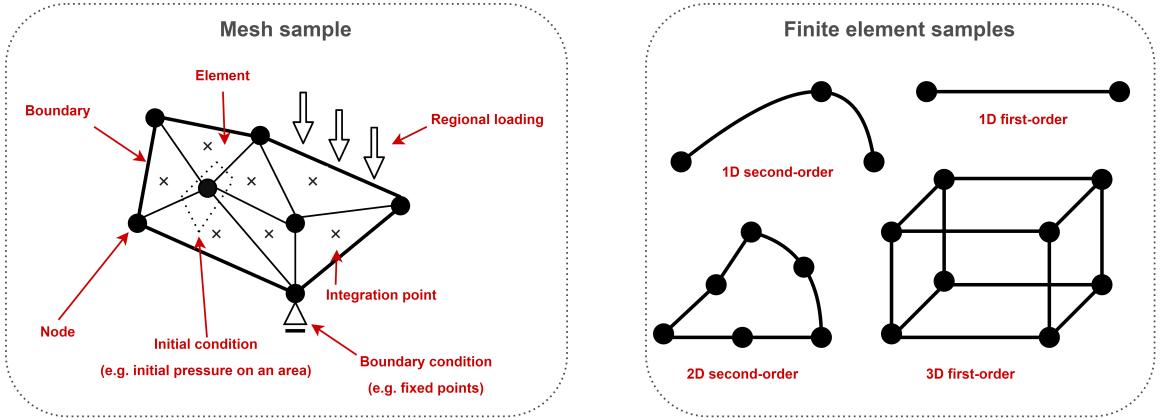


Figure 2.3: Examples of finite elements and mesh entities: typically, the geometric domain is discretized into finite elements, and the boundary conditions are applied to the nodes. In this manner, integrals are computed at integration points and are then usually extrapolated to the nodes. The choice of element definition depends on dimensionality and physical characteristics, such as geometric complexity.

$$\mathcal{F} = \left[ \begin{array}{l} \left[ \int_V \mathcal{N}^u \nabla_x \sigma \, dV - \int_S \mathbf{T} \mathcal{N}^u \, dS \right]_{t+\Delta t} \\ \int_V \left( \mathcal{N}_{t+\Delta t}^p \left( [J\rho\phi^F]_{t+\Delta t} - [J\rho\phi^F]_t \right) - \Delta t \left[ \rho\phi^F \nabla_x \mathcal{N}^p \cdot \mathbf{v}_r \right]_{t+\Delta t} \right) \, dV \\ \hookrightarrow +\Delta t \int_S \left[ \mathcal{N}^p \frac{\rho}{\mu} \mathbf{n} \cdot \mathbf{K} \cdot \nabla_x \mathcal{N}^p \bar{P} \right]_{t+\Delta t} \, dS \end{array} \right] = \mathbf{0}. \quad (2.35)$$

Since both of the shape functions are geometrical functions, their gradients can be determined. Therefore, the governing equations are fully discretized, and all derivatives are approximated. The final form of this equation depends on the exact formulation of the shape functions, the selected constitutive equations, the numerical integration method, and the dimensionality of the problem. Depending on the fidelity, the final form can be different, for instance, when viscosity is used instead of the fluid pressure, or when contact equations are also added [Pore et al. 2021; Orava et al. 2022; Orozco et al. 2022; Ardatov et al. 2023]. Therefore, eq. 2.35 is merely an illustrative example, and as we used Abaqus 2021 here, interested readers might refer to its documentation for more specific equations.

This approach is termed implicit FE analysis, as it yields equations connecting unknown parameters at  $t + \Delta t$  with known values at time  $t$  after discretization. By the way, it remains a set of nonlinear equations that first need to be linearized and then solved, for example, by Newton's method [Almeida et al. 1997; Belytschko et al. 2014]. In this regard, the system's variables, in this case,  $\{u_1, u_2, u_3, P\}$ , are denoted by  $\mathfrak{U}$ . In the 3D Cartesian system,  $u_1, u_2$ , and  $u_3$  are the  $\mathbf{u}$  components. To apply Newton's method for linearizing eq. 2.35, represented by  $\mathcal{F}$ , the Jacobian matrix  $\mathfrak{J}^{(i)}$  is used during each

iteration step  $i$ . This matrix is formulated as

$$\mathfrak{J}^{(i)} := \begin{bmatrix} \frac{\partial \mathcal{F}_1^{(i)}}{\partial u_1^{(i)}} & \dots & \frac{\partial \mathcal{F}_1^{(i)}}{\partial P^{(i)}} \\ \frac{\partial \mathcal{F}_2^{(i)}}{\partial u_1^{(i)}} & \dots & \frac{\partial \mathcal{F}_2^{(i)}}{\partial P^{(i)}} \end{bmatrix}, \quad (2.36)$$

with

$$\mathcal{F}^{(i)} := \mathcal{F} \Big|_{\mathfrak{U}^t = \mathfrak{U}^{(i-1)}, \mathfrak{U}^{t+\Delta t} = \mathfrak{U}^{(i)}}. \quad (2.37)$$

Let  $\mathfrak{U}^{(0)}$  be the known value of  $\mathfrak{U}$  determined at the start of the increment, using  $\mathfrak{U}^{(1)}$  as the initial guess, the nonlinear equations are linearized sequentially as:

$$\mathcal{F}^{(i+1)} := \mathcal{F}^{(i)} + (\mathfrak{U}^{(i+1)} - \mathfrak{U}^{(i)}) \cdot \mathfrak{J}^{(i)}. \quad (2.38)$$

These linear equations can then be solved using any standard linear algebra techniques to determine  $\mathfrak{U}^{(i+1)}$ . Convergence is confirmed when  $\mathcal{F}^{(i+1)}$  approaches zero, indicating that the unknowns are well-approximated [Belytschko et al. 2014].

A user of Abaqus can define numerical problems, such as boundary conditions, geometry, constitutive behavior, element formulation, and more. The software implements eq. 2.38 for the specified problem automatically, and its integration with Python scripts, simplifies numerical implementation. However, implementing a multiphasic model is an exception. Although Abaqus can approximate biphasic behavior using the soil consolidation theory [Verruijt 1984], Fortran subroutines are necessary for implementing multiphasic equations. In these subroutines, eq. 2.16 is directly coded along with its contribution to the Jacobian matrix (eq. 2.36). This can be approximated using a generic finite difference algorithm [Miehe 1996]. Interested readers might refer to previous tutorials [Nolan et al. 2022; Fehervary et al. 2020] for more details.

Such numeral methods, while could be expensive, still might systematically generate training data for possibly efficient surrogate modeling. These ML-based algorithms are commonly implemented using deep learning, as explained in the following.

## 2.3 Data-Driven Modeling

### 2.3.1 Deep Learning

In supervised machine learning, the central task involves designing a model function  $f$ , mapping each element in a set of input data  $\mathcal{X}$  to a corresponding element in an output set  $\mathcal{Y}$ . This relationship is mathematically expressed as  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . The elements in  $\mathcal{Y}$ , also known as labels, are derived from either human or automated supervision, and are used to develop and assess the model. These labels, in conjunction with input data,

are characterized by multidimensional features. For instance, if the values of boundary conditions in a simulation can be seen as the input features, the labels would be the observed or calculated deformation in the tissue.

Deep learning algorithms, like many of the other supervised machine learning methods, seek the most effective function  $f$  using a group of labeled data  $\mathcal{Z} \subset \mathcal{X} \times \mathcal{Y}$  and applying an optimization algorithm. This is achieved through minimizing errors, quantified by a loss function  $\mathcal{L} : \mathcal{M}(\mathcal{X}, \mathcal{Y}) \times \mathcal{Z} \rightarrow \mathbb{R}$ , where  $\mathcal{M}$  denotes the set of all potential learnable functions, which can have various hypothetical definitions like different artificial neural networks [Abiodun et al. 2019].

A simple network is the multi-layer feed-forward neural network (FFNN), which links a sequence of input features through dense (fully connected) layers to subsequent layers until the final output layer is reached [Rumelhart et al. 1986]. For an FFNN model  $\Psi \in \mathcal{M}(\mathcal{X}, \mathcal{Y})$  with  $L$  layers, the function can be expressed as:

$$\Psi(x; \cdot, \Theta) = \psi^{(L)} \left( \psi^{(L-1)} \left( \dots \left( \psi^{(1)} \left( \psi^{(0)}(x) \right) \right) \right) \right). \quad (2.39)$$

In this model,  $\psi^{(i)}$  represents the transformation through layer  $i$ , with 0 being the input layer, and  $\Theta$  includes the set of trainable parameters to be determined by the optimization algorithm. The simplest form of  $\psi^{(i)}$  for regression problems is:

$$\psi^{(i)}(x) = \begin{cases} W^{(L)}\psi^{(L-1)}(x) + b^{(L)}, & \text{if } i = L, \\ a(W^{(i)}\psi^{(i-1)}(x) + b^{(i)}), & \text{otherwise,} \end{cases} \quad (2.40)$$

where  $W$  and  $b$  are the vectors of weight and bias parameters, and  $a(\cdot)$  is the activation function, which introduces nonlinearity to capture complex nonlinear patterns [Berner et al. 2022]. Accordingly, a neuron, which is the most basic computational unit, is each member of  $\psi^{(i)}$ .

A common activation function, used in eq. 2.40, is the rectified linear unit (ReLU) [Fukushima 1980; Nair et al. 2010], defined as:

$$\text{ReLU}(x) = \max(0, x). \quad (2.41)$$

This piecewise linear function only outputs the positive input signal that mimics the threshold-based firing in biological neurons. Due to the simplicity of this function, it has many optimized implementations, and therefore it is highly used. However, it can encounter the "dying ReLU" issue during optimization, where the neurons might become too inactive. A frequently used alternative is the exponential linear unit (ELU) which

defines a small exponential function for the negative signal, as follows :

$$\text{ELU}(x; \lambda^{\text{ELU}}) = \begin{cases} x & \text{if } x > 0, \\ \lambda^{\text{ELU}}(e^x - 1) & \text{if } x \leq 0, \end{cases} \quad (2.42)$$

where  $\lambda^{\text{ELU}}$  is a hyperparameter, meaning it is a parameter that cannot be learned and must be specified by the user to define the network architecture. During the training process, an initial guess is made for the learnable parameters ( $\Theta$ ), which are then iteratively adjusted by the optimizer based on the parameter gradient. While it is possible to adjust hyperparameters using another optimizer, this approach, known as hyperparameter search or tuning, can be overly time-consuming. Therefore, especially in the context of this study, hyperparameters are usually set by the user through trial and error, especially by considering commonly applied settings.

The input during training consists of batches of concatenated samples. A large batch size can accelerate training by processing larger chunks of data, while a smaller batch size introduces noise into the training process. This noise can help the optimizer escape suboptimal local minima. Additionally, deep neural networks often encounter the issue of vanishing or exploding gradients, stemming from inconsistent scaling of network signal values. A common modification to eq. 2.39 involves using normalization layers either after or before each hidden layer, typically applied on either the batch or feature dimension. Although the input layer is usually normalized for the same reason, the output layer is sometimes normalized as well, especially if the features are of different scales. This approach enables more stable training and convergence [Ioffe et al. 2015; Bjorck et al. 2018; J. Xu et al. 2019; F. Gao et al. 2020; Singh et al. 2022].

In addition to normalization, the method used for parameter initialization also determines the speed of convergence. These methods sample parameters for each layer in proportion to the number of input or output signals so that each layer with a higher number of signals has a smaller variance range. Thus, during training initialization, the network may maintain a consistent layer-wise variance of activations and gradients across layers. The Xavier initialization technique [Glorot et al. 2010] considers both input and output signals, whereas the He initialization [k. He et al. 2015] assumes the number of output signals to be the same as the inputs, which might be more appropriate with certain activation functions [Géron 2019].

Alternatively, the LeCun initialization [LeCun et al. 2012] only considers the input signals, which may be more suitable for the scaled ELU (SELU) activation function [Klambauer et al. 2017; Géron 2019]. This function is a scaled version of the ELU function that empirically ensures self-normalization, thus potentially obviating the need for normaliza-

tion layers. It is defined as follows:

$$\text{SELU}(x) \approx 1.0507 \begin{cases} x & \text{if } x > 0, \\ 1.6733(e^x - 1) & \text{if } x \leq 0. \end{cases} \quad (2.43)$$

Regardless of the selected training settings and network architecture, once the parameters are initialized, optimizers adjust them by:

$$\theta_{t+1} = \theta_t - \alpha U_t, \quad (2.44)$$

where  $\alpha$  is the learning rate hyperparameter and  $U_t$  is the update function. In the context of deep learning, where high-order optimizers could be ill-conditioned due to the noise in each mini-batch of labeled data, Adam, a popular and efficient first-order optimizer, is used. This optimizer is an extension and combination of previous first-order optimization methods [Kiefer et al. 1952; Polyak 1964; Nesterov 1983; Duchi et al. 2011; G. Hinton et al. 2012], defined by:

$$U_t = \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \quad (2.45)$$

with  $\epsilon$  being a small constant to prevent zero-division. Also,  $\hat{m}_t$  and  $\hat{v}_t$  are respectively the modified first and second moments, defined by:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (2.46)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}, \quad (2.47)$$

where  $\beta_1$  and  $\beta_2$  are typically hyperparameters close to one.  $m$  and  $v$  are respectively the mean and variance estimates, updated by:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\Theta} \mathcal{L}(\Theta_t) \quad (2.48)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\Theta} \mathcal{L}(\Theta_t))^2. \quad (2.49)$$

Conceptually, since these metrics are initially zeroed, they are biased toward small values in the early iterations, and therefore their values are augmented by eqs. 2.46 & 2.47. The hyperparameters in eqs. 2.48 & 2.49 help in smoothing out the updates to dampen oscillations. These accumulations of gradient and its standard deviation are then applied into eq. 2.45 so that it adaptively scales down the gradients with huge variances to avoid overshooting the global minimum.

To implement eq. 2.44, a loss function should be selected to measure prediction error. Since our deep learning problems here are regression tasks, where even the outliers (like irregular numerical noises) are significant, we used the  $l_2$  loss, known as the mean squared

error (MSE), which is basically the Euclidean distance between the prediction vector and target vector. This differentiable function can now be differentiated, and then, using the chain rule of calculus, it can be related to the gradient of each parameter, i.e.,  $\nabla_{\Theta}\mathcal{L}(\Theta_t)$ . This process is known as the backpropagation algorithm [Rumelhart et al. 1986], where the gradient is calculated in a backward manner, from the output to the input layers, typically using an automatic differentiation algorithm; see the [Baydin et al. 2018] study for further details.

One of the key aspects of training involves the training loop, where a batch of input data is fed into the model, and the parameters are updated based on the loss function and optimization algorithm. This loop repeats until all data in the training set has been processed, marking the completion of an epoch. Metrics such as MSE are used to assess model performance on different data splits (training, validation, and testing sets), which helps in evaluating the model’s generalizability to unseen data. The distinction between in-distribution and out-of-distribution testing errors is made based on the similarity of the testing set to the training set. Additionally, to prevent overfitting, an early stopping mechanism is often employed where the training process is halted if the validation error starts increasing. This ensures that the model retains its generalization ability and does not become overly complex for the given data [Géron 2019].

In practical terms, deep learning models especially for regression tasks, such as those described in this study, are typically implemented using high-level libraries like Keras [Chollet et al. 2015] or Sonnet [Reynolds et al. 2017], together with TensorFlow [Abadi et al. 2015] and other Python tools [Hunter 2007; Pedregosa et al. 2011; K. et al. 2016; Harris et al. 2020; Virtanen et al. 2020]. These libraries provide a rich set of tools, including built-in network elements, data handling utilities, and automatic implementation of training optimizers and backpropagation. This simplifies the process of setting up and training a model, as most of the complex underlying mechanisms are abstractly handled by the libraries. The user’s primary task is to define the network architecture, after which these tools take care of the iterative training process. Also, although these tools offer utilities to help manage overfitting, we chose simplified models for faster training in this study and did not use them. Our main challenge is indeed to prevent underfitting, which can lead to suboptimal performance.

On the other hand, deep learning can also be applied in an unsupervised, or more precisely, self-supervised manner, where the input and output data are the same. These so-called autoencoders (AEs) are designed to learn typically reduced representations (encodings), which are then reconstructed by a decoder and implemented using regular FFNN. The primary use case is for dimensionality reduction. However, a notable variation is the denoising AE, which involves adding artificial noise to the data in a preprocessing step (often referred to as DA since it generates new samples). The DAE then learns to remove this noise [G.E. Hinton et al. 2006; Vincent et al. 2008].

Another deep learning model used in this study is the message-passing graph neural networks (GNNs), which allow for permutation invariance, meaning that the input data can be fed in different orders. In such GNNs, the focus is on local behavior, particularly on a graph node and its neighbors. This updates the nodal representations as follows:

$$\mathfrak{v}'_n = \text{FFNN} \left( \sum_{i \in \Gamma(n)} \mathfrak{v}_i, \mathfrak{v}_n \right), \quad (2.50)$$

where  $\mathfrak{v}'_n$  is the updated nodal feature. The effectiveness of the model can be further enhanced by incorporating edge features and updating them with another FFNN [H. Cai et al. 2018; J. Zhou et al. 2020; Z. Wu et al. 2021]. The main application here is in scenarios where geometrical generalizability is critical. This is achieved by first learning local behavior and then generalizing to other geometries.

In summary, the selection of a DL method depends largely on the objectives at hand, with a focus on employing such learnable AI systems instead of relying on predefined physical models. This approach, particularly in the current era dominated by Python and its extensive libraries, becomes increasingly feasible given the availability of sufficient data and computational resources.

### 2.3.2 Surrogate Modeling

Surrogate methodologies offer a synergy of data-driven strategies and computational analytics, providing efficient simulators widely applied in diverse fields such as healthcare [Holzapfel et al. 2021; Shim et al. 2020; L. Liang et al. 2018] and engineering [Pfaff et al. 2021; W. Gao et al. 2020; Kalyuzhnyuk et al. 2019]. These methods, particularly deep learning or alternative regression techniques, are adept at identifying patterns within FE model samples. As both physics-based and AI methods are relatively easy to implement, surrogate models are also straightforward to apply, especially for simple models.

As an illustrative example (Fig. 2.4), a basic FE model is formulated in Abaqus to examine contact mechanics in a 2D rectangular tissue with an embedded circular hole to complicate mesh organization. This model employs a straightforward monophasic framework (i.e., an incompressible neo-Hookean formulation). Simulation results are obtained for each mesh point, from the vertical Cauchy stress. An FFNN is then employed to learn the physics on a regular computer. This open-sourced<sup>1</sup> surrogate model mirrors those found in standard cartilage research but with further simplifications for clarity [Paiva et al. 2012; Arbabi et al. 2016a; Arbabi et al. 2016b]. To expedite the learning process, the model's complexity is reduced by incorporating a single hidden layer that contains a mere fraction of the neurons compared to the final layer. This approach parallels the concept

---

<sup>1</sup><https://github.com/shayansss/pmse>.

of reduced-order models for fast training [Pant et al. 2021], albeit utilizing an FFNN.

For the evaluation of the surrogates, global metrics, such as the MSE, are widely used [Phellan et al. 2021]. For a surrogate, for instance, we assume that it can predict the outputs of the numerical model extracted at  $N$  assigned points (such as FE nodes<sup>1</sup>) evaluated on  $M$  test samples. Accordingly,

$$\text{MSE} = \frac{1}{M} \sum_{m=1}^M \text{SE}_m \quad \text{with} \quad \text{SE}_m = \frac{1}{N} \sum_{n=1}^N (\bar{y}_{m,n} - y_{m,n})^2. \quad (2.51)$$

Here,  $\bar{y}_{m,n}$  and  $y_{m,n}$  are the prediction and target values of the  $n^{\text{th}}$  point of sample  $m$ , respectively, and  $\text{SE}_m$  is the averaged squared error corresponding to sample  $m$ . Despite the benefits of these popular global or averaged metrics, they might be barely interpretable beyond performance comparison [Molnar 2019]. Therefore, it is common practice to visualize the prediction for each sample [Sanchez-Gonzalez et al. 2020]. However, this approach focuses only on each sample separately, which might bias our inference towards those selected samples. A better alternative is to use local metrics that enable pointwise visualization of the errors [Giselle Fernández-Godino et al. 2019], such as the pointwise MSE (PMSE) on point  $n$ , i.e.,

$$\text{PMSE}_n = \frac{1}{M} \sum_{m=1}^M (\bar{y}_{m,n} - y_{m,n})^2. \quad (2.52)$$

This metric allows us to visualize a contour plot of errors on the physical system, helping us to see the direct correlation between the machine learning and numerical mesh, if relevant. For other performance comparisons, MSE is preferred.

For very simplified models, as exemplified above, a surrogate may be generated efficiently. However, for complex multi-physics equations of cartilage, especially at a large scale, having a large training dataset for machine learning is not feasible. If the primary purpose of applying ML is to enhance efficiency, the models should ideally be trained on a limited number of samples, especially when the numerical generation of these samples is costly [Forrester et al. 2008]. This factor often leads to the underutilization of AI-enhanced modeling techniques. This research gap — the development of an advanced yet efficient FE analysis and ML synergy for cartilage surrogate modeling — will be addressed in the upcoming chapters. It involves advancements in both the numerical aspects and the machine learning components of the cartilage surrogate model.

---

<sup>1</sup>In practice, for most analyses, obtaining results directly from the integration points is more accurate than using extrapolated values at nodes. However, we primarily use nodal values as they better fit our graph representation learning in the next chapters. Additionally, it is necessary to perform several mesh sensitivity analyses, but a denser mesh can significantly extend data generation time, making experimentation impractical. Therefore, we adhere to the mesh quality used in previous studies. With these settings, we find a trade-off between experimental accuracy and efficiency.

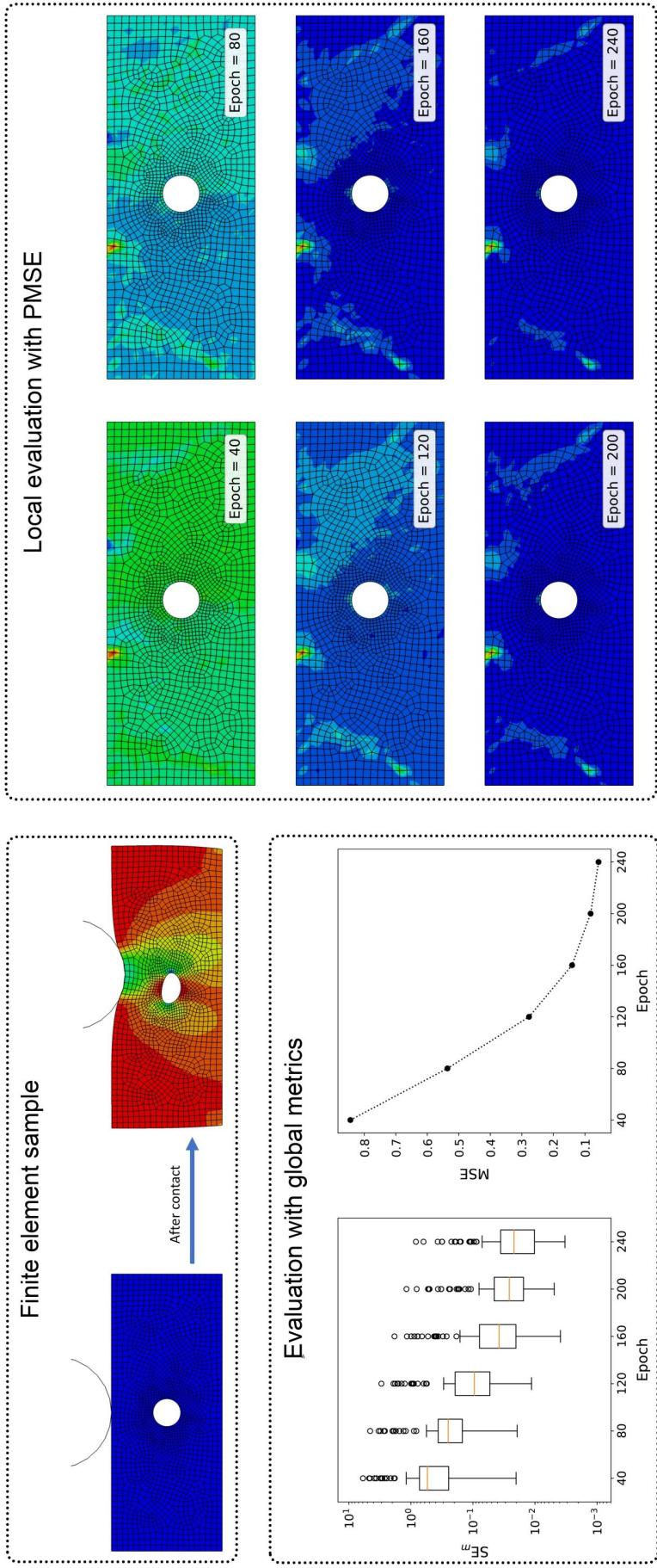


Figure 2.4: Example of a surrogate model and its evaluation. The normalized vertical Cauchy stress is assigned as the target values (as shown in the left contour subplot) and therefore is used to calculate errors. The evaluation of training performance is depicted using global metrics, with an emphasis on the widely recognized mean-squared error (MSE) metric. The  $SE_m$  box plot highlights sample anomalies as well. The point-wise MSE (PMSE) contours are superimposed on the initial configuration. Observing the final snapshots of PMSEs indicates the inaccuracy in learning where the element shapes varied. PMSE offers valuable insights for further interpretability, yet MSE remains the conventional standard for metric comparison. For contour plots, the values are shown by a color range from blue to red, corresponding to -2750 to 165 MPa for stresses and 0 to 2 for PMSEs, respectively.

# Chapter 3

## Finite Element Analysis with Pre-stressing Optimizers

This study focuses on surrogate modeling of the multi-physics cartilage FE model, a critical aspect of which is using an accurate and efficient simulator to generate datasets. This chapter explores the complexities of FE simulation, highlighting a significant challenge: the pre-stress conditions in the solid matrix that are physiologically present. We implement, compare, and validate various pre-stressing algorithms proven effective in cartilage multi-physics modeling. These algorithms employ efficient optimization techniques to determine the initial, unknown *in vivo* conditions—those conditions without pre-stressing—crucial for a HF cartilage simulator, which will be used in subsequent chapters. The algorithm is initially applied to 2D test cases and then expanded to a 3D simulation test.

### 3.1 Related Work

**Pre-stressing effects.** As explained in Chapter 2, AC is generally considered a fibril-reinforced mixture within a charged medium, potentially leading to osmotic pressure. This pressure often serves as the primary load-bearing component in AC, as demonstrated in both healthy [Quiroga et al. 2017] and damaged [Sajjadinia et al. 2019] models; therefore, it should not be overlooked in multi-physics analyses. The initial value of this pressure can be established by considering the initial condition (no deformation), i.e., volume ratio  $J = 1$ , in eq. 2.17, yielding  $\boldsymbol{\sigma}_0^{\text{GAG}} = \alpha_1 \mathbf{I}$ , with  $\alpha_1$  and  $\mathbf{I}$  being respectively a material constant and the identity tensor. This *in vivo* initial pressure induces stress in the otherwise stress-free state of the tissue; thus, it is termed the cartilage pre-stress  $\boldsymbol{\sigma}_0$ . This can be calculated by applying the initial conditions of a stress-free state (without any external load or deformation), i.e., initial deformation gradient  $\mathbf{F}_0 = \mathbf{I}$  and initial fluid pressure  $P_0 = 0$ , into eq. 2.15, resulting in initial stress  $\boldsymbol{\sigma}_0 = -\alpha_1 \mathbf{I}$ .

The FE code primarily implements pre-stressing by assuming a stress-free state as the initial condition. Introducing this swelling stress into the FE mesh naturally causes it to expand, stretching the fibers until reaching equilibrium, i.e., the pre-stressed state. The biomechanical effect of this pre-stressing-induced swelling has been studied experimentally [Narmoneva et al. 1999] and mathematically [Setton et al. 1996], recording evidence of heterogeneous residual strain field in the pre-stressed tissue. Also, the numerical comparison of fibril-reinforced multi-physics models of cartilage with and without pre-stress reveals significantly different distributions of Cauchy stress in a 2D study [X. Wang et al. 2018]. Nonetheless, this comparison only includes the effects of the pre-stressing on a small-scale geometrical configuration and therefore in this chapter, it is also extended to a larger-scale 3D model.

**Pre-stressing implementation.** The inclusion of cartilage pre-stressing in common FE code is challenging, as the existing literature predominantly records *in vivo* data with a presumption that the pre-stressed condition is the starting point. This is sharply different from most numerical models, where the stress-free state is assumed as the initial condition. This fundamental difference leads to significant inconsistencies between the starting points in numerical models and actual *in vivo* conditions [X. Wang et al. 2018]. A simplistic solution is to begin the regular numerical analysis with the experimentally determined pre-stressed configuration. However, as stated above, this approach adversely impacts the fidelity of the biomechanical simulation.

An alternative method involves trying various stress-free geometries and manually applying pre-stressing to determine which stress-free configuration most accurately aligns with the experimental configuration after pre-stressing deformation [Sajjadinia et al. 2019; Stender et al. 2017]. This method, though, is highly inefficient for repetitive tasks such as surrogate modeling, where each case requires a new round of trial and error. Therefore, a pre-stressing algorithm (PSA) is essential for automatically incorporating pre-stressing in numerical modeling.

The iterative pre-stressing or pre-straining algorithms have been applied across different scientific disciplines, e.g., in the inverse motion problem with finite hyper-elasticity [Govindjee et al. 1996] and the modeling of aortic root and simulation of the stent deployment [Caimi et al. 2020; Votta et al. 2017]. A common PSA technique is based on the multiplicative decomposition of deformation gradients [Gee et al. 2010; Weisbecker et al. 2014; Pierce et al. 2015; Zahn et al. 2016]. While effective, these methods often necessitate modifications to FE elements or constitutive equations, which might have complex implementations in many FE software.

Mathematical calculation of the pre-stressing is also possible with very specific assumptions on geometry and material properties [Alastrué et al. 2007; Rachev et al. 2003; Taber et al. 1996]. These methods are too restrictive for complex cartilage problems. An

additional approach involves fixed-point-based backward schemes (FPBBSs), known for their efficiency in finding stress-free geometry. These algorithms treat numerical models as closed systems, relying solely on the input and output geometrical values of these systems, without needing to alter the underlying numerical equations [Leach et al. 2019].

In a FPBBS, a function, here pre-stressing transformation, is defined so that a state like the stress-free configuration is its fixed point (or objective). Then, starting from an initial estimate, this function is applied, for example, by a forward (regular) FE analysis. In each iteration, the function output becomes its next input, gradually moving closer to its fixed point, while its error is measured by the forward analyses. The term *backward displacement* refers to, for example, implementing a backward (auxiliary) FE analysis to apply a displacement field on the model to reverse the effect of the pre-stressing, and getting the configuration for the next step [Leach et al. 2019].

A noteworthy PSA based on the FPBBS was developed by X. Wang et al. 2018 and implemented in a complex AC model. Despite being a significant advancement, it was primarily focused on geometry and did not account for constitutive variations arising from kinematic changes in mixture models. This limitation is addressed by introducing a new FPBBS that includes material optimization. This approach optimizes not only the stress-free states for geometric configurations but also the affected material parameters.

## 3.2 Methodology

In the following, the term (REF) in superscript denotes the experimentally verified values that the FE model aims to reach once it is under pre-stress and attains a state of equilibrium. The subscript 0 is used to denote the initial state in an analysis. Additionally, the superscript ( $t$ ) indicates the values at the  $t^{th}$  step in the PSA, which estimates the stress-free state of the involved parameters.

### 3.2.1 Pre-Stressing Optimizers

In most FE implementations, a forward analysis initiates from a state where no stress is present. Introducing pre-stress in an AC model leads to the deformation of its matrix, culminating in a new configuration. In this equilibrium state, the model's boundary conditions and solid matrix counteract the internal pressure. As the material parameters are subject to change with this new configuration, applying the initial *in vivo* values at the beginning of the simulation can lead to inaccuracies in the starting conditions for subsequent simulations. These initial *in vivo* constitutive parameters can be sourced from existing literature, as depicted in Fig. 2.2 and the equations below [Wilson et al. 2007;

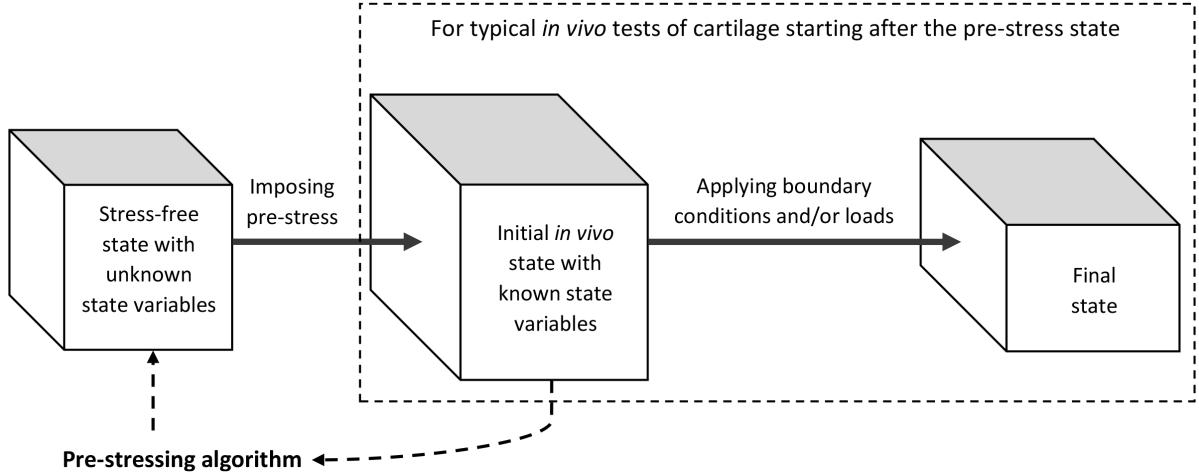


Figure 3.1: Role of the pre-stressing algorithm (PSA) in various simulation states: initially, the problem is presumed to start in a stress-free state with unknown state variables, encompassing geometrical configuration, material parameters, or both. Following the application of pre-stress, these state variables are updated to their revised values, aligning with the initial conditions of standard *in vivo* or *in vitro* tests that typically begin in a pre-stressed state. To ascertain the stress-free state, a PSA is employed, iteratively refining the initial estimates of state variables to align with the initial *in vivo* values. Once determined, the primary numerical analysis can proceed to simulate the experimental test.

Rieppo et al. 2004; Shapiro et al. 2001; Lipshitz et al. 1975]:

$$\phi_0^{S(\text{REF})} = 0.1 + 0.2z, \quad (3.1)$$

$$\rho_0^{\text{COL}(\text{REF})} = 1.4z^2 - 1.1z + 0.59, \quad (3.2)$$

where  $z$  represents the normalized depth,  $\phi_0^S$  is the initial volume fraction of the solid component, and  $\rho_0^{\text{COL}}$  is the initial value of the total collagen volume fractions. The *in vivo* data utilized in this study are averages of experimental data, thus excluding outliers.

Regarding the PSA [Bols et al. 2013], the initial *in vivo* (pre-stressed) state and stress-free state are considered known and unknown variables, respectively. In the beginning, a forward analysis begins with the *in vivo* state as the first approximate stress-free state. Subsequently, the stress-free state is adjusted using an update function  $\mathbf{U}$ , and the forward analysis is repeated until the residual function  $r$ 's value converges to a negligible amount. This suggests that the initial pre-stressed state is closely aligned with the *in vivo* observations. This technique is capable of optimizing either the geometrical configuration alone (as implemented by X. Wang et al. 2018), or both geometrical and material parameters, as demonstrated in this work (Fig. 3.1).

Now, considering the initial values of the parameters influenced by pre-stressing in our numerical model, symbolized by  $\mathbf{s}$ , they encompass geometrical data  $\mathbf{x}$  and constitutive data  $\mathbf{m}$  (components include fibrillar direction  $\mathbf{n}$ , solid volume fractions  $\emptyset^S$ , and total

collagen volume fractions  $\rho^{\text{COL}}$ ). Our PSA conducts the following sequence of analyses:

$$\mathbf{s}_0^{(t)} \xrightarrow{\text{forward analysis}} \mathbf{x}_0^{(t+1)} \xrightarrow{\text{backward analysis}} \mathbf{m}_0^{(t+1)}. \quad (3.3)$$

The forward analysis aims to estimate the stress-free geometry, and the backward analysis determines the alterations in material properties at this state.

**Forward analysis.** When pre-stressing is applied to the model in its initial state  $\mathbf{s}_0^{(t)}$ , the pre-stressed configuration  $\hat{\mathbf{x}}^{(t+1)}$  is derived through an FE analysis and the multiphasics constitutive model eqs. 2.16 & 2.15. The update function  $\mathbf{u}$  is defined as

$$\mathbf{u}^{(t)} := \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(\text{REF})}, \quad (3.4)$$

and the stress-free geometry is edited in the following manner:

$$\mathbf{x}_0^{(t+1)} := \begin{cases} \mathbf{x}_0^{(t)} - \zeta^{(t)} \mathbf{u}^{(t)} & \text{if } r^{(t)} \leq r^{(t-1)}, \\ \mathbf{x}_0^{(t)} & \text{if } r^{(t)} > r^{(t-1)}, \end{cases} \quad (3.5)$$

with  $r$  and  $\zeta$  being respectively the residual and scaling factor, each determined by the optimization algorithm.

**Backward analysis.** The PSA, now, confines all the AC nodes with the (displacement-based) boundary condition  $\mathbf{b}$ , using

$$\mathbf{b}^{(t+1)} := \begin{cases} -\zeta^{(t)} \mathbf{u}^{(t)} & \text{if } r^{(t)} \leq r^{(t-1)}, \\ \mathbf{0} & \text{if } r^{(t)} > r^{(t-1)}. \end{cases} \quad (3.6)$$

This allows another (backward) FE analysis to compute the new constitutive state  $\mathbf{m}_0^{(t+1)}$  using eqs. 2.20 and 2.19, starting with state  $\mathbf{s}_0^{(t)}$ :

$$\mathbf{n}_0^{(t+1)} := \frac{\check{\mathbf{F}}^{(t)} \mathbf{n}_0^{(t)}}{\left\| \check{\mathbf{F}}^{(t)} \mathbf{n}_0^{(t)} \right\|_2}, \quad (3.7)$$

$$\varphi_0^{(t+1)} := \frac{\varphi_0^{(t)}}{\det \check{\mathbf{F}}^{(t)}} \quad \text{with } \forall \varphi \in \{\emptyset^S, \rho^{\text{COL}}\}. \quad (3.8)$$

In this context,  $\check{\mathbf{F}}$  represents the deformation gradient in the inverse analysis, as determined by the computational solver.

**Optimization.** With the following initial values,

$$\begin{cases} \mathbf{s}_0^{(0)} := \mathbf{s}^{(\text{REF})}, \\ \mathbf{u}^{(0)} := \mathbf{0}, \\ \zeta^{(0)} := 1, \\ r^{(0)} := \infty, \\ \mathbf{b}^{(0)} := \mathbf{0}, \\ \check{\mathbf{F}}^{(0)} := \mathbf{I}, \end{cases} \quad (3.9)$$

the above-mentioned optimizer allows estimation of the new initial state  $\mathbf{s}_0^{(t+1)}$ , i.e.,  $\mathbf{x}_0^{(t+1)}$  and  $\mathbf{m}_0^{(t+1)}$ . This process necessitates the use of the following equations:

$$r^{(t)} := \|\mathbf{U}^{(t)}\|_2, \quad (3.10)$$

$$\zeta^{(t)} := \begin{cases} \zeta^{(t-1)} & \text{if } r^{(t)} \leq r^{(t-1)}, \\ \frac{\zeta^{(t-1)}}{\eta} & \text{if } r^{(t)} > r^{(t-1)}, \end{cases} \quad (3.11)$$

where  $\mathbf{U}$  encompasses all components of  $\mathbf{u}$  at every node, and  $\eta = 4$  acts as a hyper-parameter for averting potential divergence (adjustable for better convergence). The optimization is assumed to be converged when the error derived from eq. 3.10 diminishes significantly, signifying the approximation of the stress-free state  $\mathbf{s}_0$  is achieved through our FPBBS. For clarification, a schematic of convergence is illustrated in Fig. 3.2.

This algorithm can be further modified by using a different residual function, such as the infinity norm, to enhance pointwise optimization. Additionally, rather than calculating volume fraction changes in backward analysis, an alternative optimizer could be employed to test various initial values, achieving a better approximation of pre-stressed values. The sequential application of constitutive and geometrical parameters may improve convergence. Despite variations in implementation, all methods adhere to the same concept of cartilage PSA with also the material optimization (MO).

### 3.2.2 Simulation Tests

**Numerical Implementation.** The FE implementation of the AC pre-stressing, as depicted in Fig. 3.3, involves forward analyses that incorporate stress-free material properties. This is achieved via our Fortran SDVINI subroutine, responsible for setting the initial state variable values at each integration point in the numerical model. This subroutine operates in alignment with Abaqus' step 0, which determines the geometric attributes (positions) of these points. Consequently, material parameters can be initialized with respect to the normalized depth  $z$ , and these computed variables are further processed

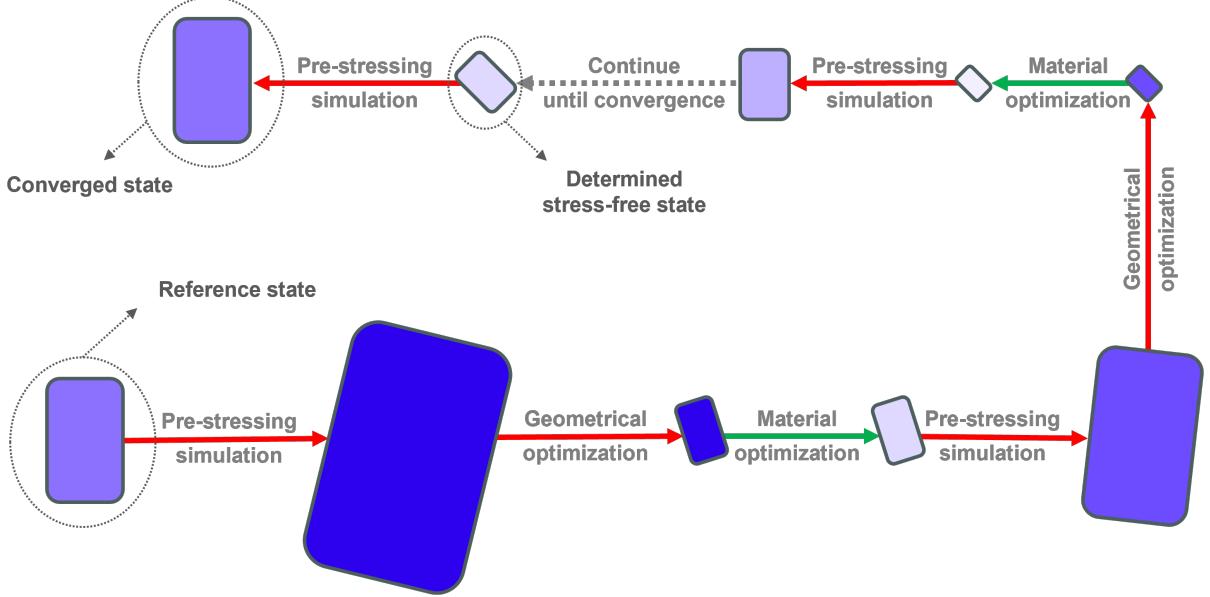


Figure 3.2: Schematic of the pre-stressing algorithm, where the material state is illustrated by a blue shade, and blue and red arrows show the forward and backward analyses, respectively. At each step, pre-stressing is applied to achieve a new configuration, which is then compared with the reference state to calculate the new geometrical configuration, while the material state is also updated afterward. It is possible to optimize the material and geometrical states with separate optimizers or using a unified optimizer.

through our Fortran UMAT subroutine for modeling the solid components of the tissue.

The FPBBS, as outlined in the preceding section, is implemented through a Python script. Initially, two FE models, mirroring each other and based on initial experimental data, are constructed for forward and inverse FE analysis. Both models are assigned the same boundary conditions, with the inverse model incorporating also displacement nodal field  $\mathbf{b}$  at each AC node, starting at zero to represent an undeformed state.

Following the backward analysis, the newly derived parameters are reintegrated into the SDVINI subroutine to initiate a fresh cycle of PSA. This iterative process persists until convergence is attained. However, substantial variations in the updating state variables during certain iterations may hinder the FE analysis or PSA convergence, which can be managed by moderating parameter updates, albeit increasing iteration counts. Figure 3.4 illustrates the PSA implementation in Abaqus, as employed in the following simulation experiments.

**2D tests.** Aligning with X. Wang et al. 2018’s earlier study on AC pre-stressing, it is assumed that pre-stress alterations predominantly affect the fibrillar component of the cartilage matrix. Accordingly, tensile tests, both with and without material updates, are simulated using fully integrated 8-node biquadratic displacement and bilinear pore pressure elements. These simulations incorporate velocity boundary conditions to emulate a 1.2 stretch over 4000 Sec (refer to Fig 3.5, right). For material calibration, a SciPy-based

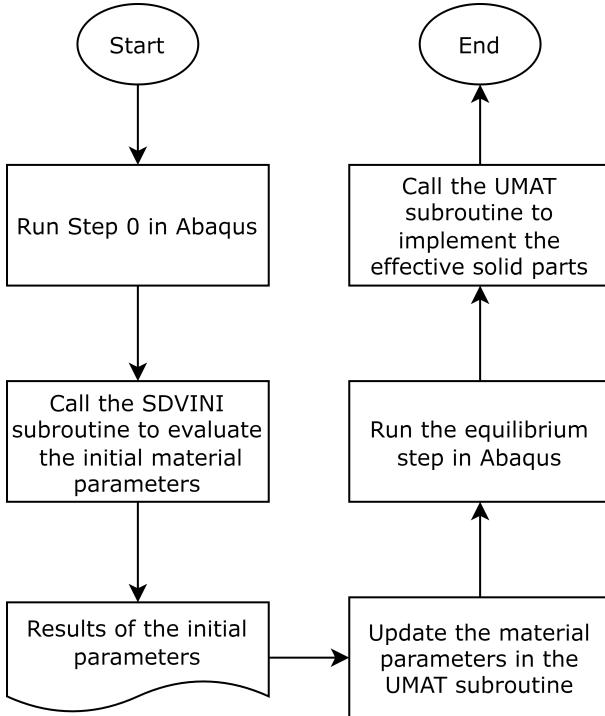


Figure 3.3: Schematic representation of the pre-stressing implementation in cartilage models, using the Fortran subroutines of Abaqus.

optimizer is employed to align FE models with varying fibrillar elasticity to empirical data [Elliott et al. 2002], using experimental stress and stretch records <sup>1</sup>. The concept of experimental biomechanical parameters during pre-stressing is detailed in X. Wang et al. 2018’s research.

In exploring the significance of including material parameters in pre-stressing updates, the PSA is conducted both with and without constitutive updates using an unconfined compression test. For them, fully integrated eight-node axisymmetric quadrilateral, bi-quadratic displacement, and bilinear pore pressure elements are selected, where the radial movement along the symmetry axis is restricted (refer to Fig 3.5, left). Then, the alterations in material fractions along the symmetry axis and the changes in primary fibril orientations are calculated. In the following compression test, a cartilage plug of 0.5 mm radius underwent axial compression up to 10% strain, guided by velocity boundary conditions on the upper nodes at  $0.002 \text{ Sec}^{-1}$ , finished by a relaxation phase.

Moreover, in all 2D simulations, the modeled cartilage has a maximum of 1 mm depth and zero fluid pressure at its outer boundaries, allowing fluid to flow freely. The nodes on the AC’s lower surface are fixed completely to mimic the osteochondral interface.

---

<sup>1</sup>Theoretically, the pre-stressing mechanism of soft tissues can be constituent-specific, and material calibration might affect both the fibrillar and non-fibrillar components [Maes et al. 2019]. However, in the case of cartilage, the osmotic pressure primarily affects the fibrillar constituents [Quiroga et al. 2017]. Therefore, calibration is limited only to the fibrillar component, and in line with the previous study [X. Wang et al. 2018], constituent-specific PSAs are disregarded.

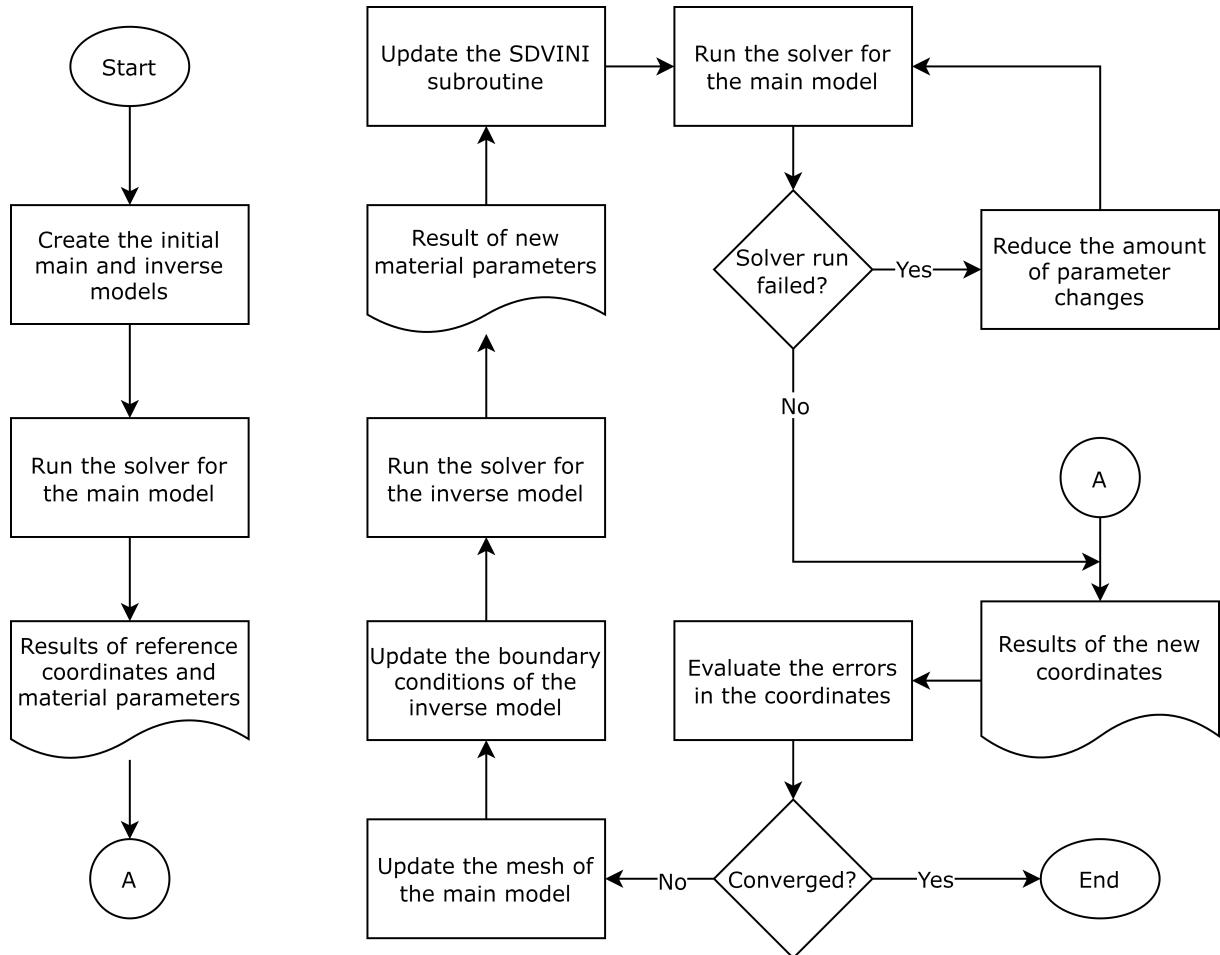


Figure 3.4: Flowchart of the backward optimization of our pre-stressing algorithm, executed in Abaqus using a Python script.

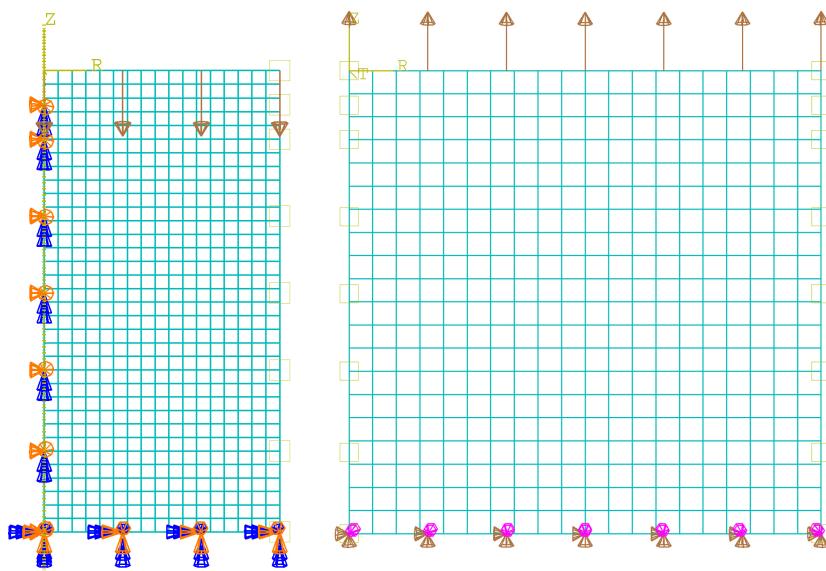


Figure 3.5: 2D models with an axisymmetric geometry (left) and a plane strain geometry (right).

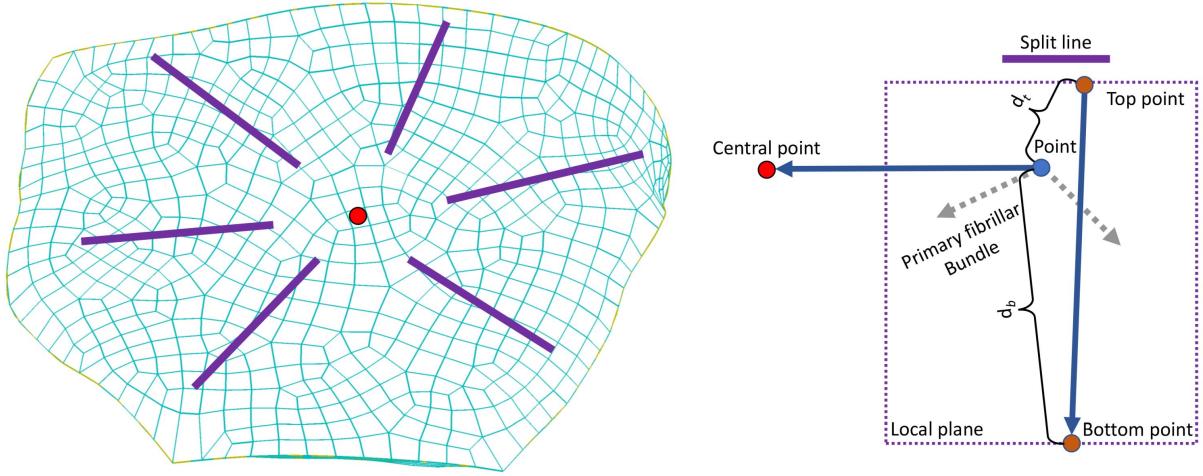


Figure 3.6: Example of the orientations of split lines of a cartilage substructure (left) and a local plane corresponding to a cartilage point (right).

**3D test.** The knee model is derived from the Open Knee Project [Erdemir 2016], based on MRI scans of a 70-year-old woman’s cadaver. Here, the 2nd-order 3D brick elements with pore pressure equations are utilized for the cartilage models. Bones are modeled with rigid elements, and other knee components are modeled using eight-node three-dimensional (3D) linear brick elements. Cartilage-bone and ligament-bone interfaces are tightly bound, while bones are subjected to full constraint. The material models used for the other substructures are extracted from the Shim et al. 2016 study.

Additionally, for this joint-scale cartilage model, the normalized depth  $z$  definition is redefined to accommodate the knee models with varying thicknesses. The position of each point is used to find the nearest nodal coordinates on the AC’s upper and lower surfaces (signified by  $d_t$  and  $d_b$ , respectively), with the aid of a nearest neighbor search algorithm [Maneewongvatana et al. 1999]. Then,

$$z = \frac{d_t}{d_t + d_b}. \quad (3.12)$$

Now, considering the model’s asymmetry, a local plane is constructed at each AC point, assuming that the primary fibrils within this plane mark the split-lines patterns on the AC surface. These lines are generally assumed to be directed toward the center of each cartilage substructure [Mononen et al. 2012]. Therefore, the local plane encompasses one vector facing the central point and another linking the nearest surface points, as shown in Fig. 3.6 with blue arrows. With this plane and the calculated normalized depth, the fibrillar orientations are defined, in the same way as the ones used in 2D models, complementing the FE implementation. The code of the large-scale model is

open-sourced<sup>1</sup> due to its complexity.

### 3.3 Results and Discussion

In this study, we introduce a novel PSA that integrates nodal geometric optimization and material optimization (MO) for FE analysis. This approach is applied to an anisotropic, depth-dependent AC mixture model, which comprises all major multiphasic components explained in the previous chapter. Our proposed solution addresses the inconsistency between actual *in vivo* states and numerically-implemented pre-stressed states. This is achieved by incorporating constitutive variables into existing FPBBSs [X. Wang et al. 2018; Bols et al. 2013; Pandolfi et al. 2008] and by defining an inverse analysis for inclusion of these variables.

The main hypothesis of this study is that the heterogeneous constitutive properties of the multiphasic AC model can be modified through the application of pre-stressing via a FPBBS with material variations, and this modification can influence the biomechanics of the tissue. To evaluate this hypothesis, a series of 2D tests was conducted, followed by a larger-scale analysis involving a tibiofemoral joint to examine the effects of pre-stressing. The proposed algorithm proved its high efficiency, as it automatically identifies the stress-free state within a maximum of six hours using a regular hardware setup in the most resource-intensive model (used in the 3D test).

**2D tensile test.** The constitutive equations, previously validated [Sajjadinia et al. 2019], were not previously combined with pre-stressing methods. Therefore, we simulated a series of tensile tests, both with and without material variation across a broad spectrum of fibrillar elasticity parameters. The initial parameter values were based on earlier research [Wilson et al. 2007]. Table 3.1 presents the derived values for both scenarios, and Fig. 3.7 compares the simulation results with experimental data [Elliott et al. 2002].

Table 3.1: Fitted constitutive parameters of the fibrillar network for different implementations. Abbreviations: MO = material optimization; PSA = pre-stressing algorithm.

Parameter	PSA with MO	PSA without MO
$E_1$ (MPa)	19.51	30.21
$E_2$ (MPa)	230.67	234.11

Both calibrated FE models exhibited close alignment with median values: initially, the stress-strain curves show mild nonlinearity, reflecting the cartilage's near-linear behavior at the onset. However, these curves slightly underestimate stress compared to median experimental values. At a stretch value of approximately 1.06, the response became dominated by the nonlinear elasticity of fibrils, leading to a minor overestimation of

<sup>1</sup><https://github.com/shayansss/psa>.

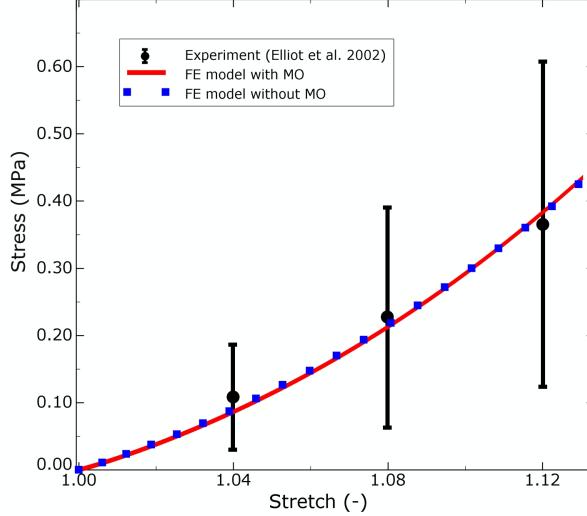


Figure 3.7: Recorded stress vs. stretch extracted from calibrated numerical models mimicking an experimental test [Elliott et al. 2002]. Abbreviations: MO = material optimization; FE: finite element.

stress. Nonetheless, both recorded responses fall within the range of experimental data, indicating their applicability in general AC modeling.

Regardless of the similarity in the results, the model without material alteration displayed slightly more strain-related fibrillar elasticity  $E_2$  but significantly higher initial elasticity  $E_1$ . This observation could be attributed to the initial pre-stressing state where the fibrillar network predominantly resists osmotic pressure [Quiroga et al. 2017]. Also, since the model without MO has decreased material fractions, as per eq. 2.22, there is a need to increase fibrillar elasticity, particularly the initial elasticity, to compensate for these reductions, especially at the small deformation.

**2D compression test.** In pursuit of validating our hypothesis, we conducted FE simulations in a confined compression scenario. These simulations were executed with previously calibrated constitutive models, both with and without optimizing the material parameters via the PSAs. Accordingly, Fig. 3.8 displays the distribution of the angle  $\theta$ , signifying the deviation between the radial axis and one of the exemplified fibrillar directions upon achieving equilibrium. The model incorporating MO exhibited precise alignment with *in vivo* observations. In contrast, the model lacking MO deviated in the peripheral regions, with angle variations between 5 and 10% compared to the *in vivo* measurements.

Figure 3.9 displays the variations in the volume fractions of the constitutive equations along the normalized depth in the cartilage's axis of symmetry. While both models demonstrated analogous trends, the non-MO model exhibited approximately 8 to 16% divergence from the experimental values across most regions; however, the deeper zones showed convergence in both models. Discrepancies in such constitutive parameters sup-

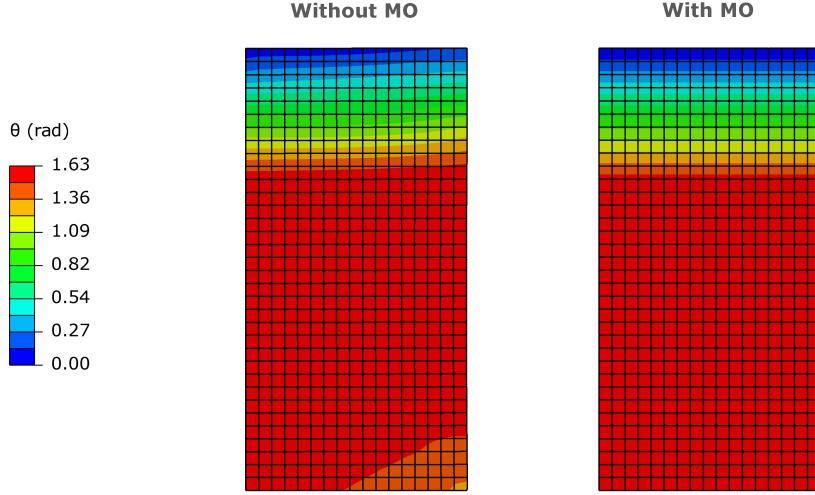


Figure 3.8: Contour of a representative fibrillar angle  $\theta$  with respect to the radial axis in the pre-stressed state. Abbreviation: MO = material optimization.

port this study's hypothesis, considering that they indicate the influence of MO on the biomechanics of cartilage. Supporting this, Julkunen et al. 2008 noted that variations in volume fractions significantly impact mechanical responses while maintaining underlying trends. These findings indirectly validate the enhanced precision and significance of MO incorporation in PSA. Further insights into the biomechanical effects are being explored through the following relaxation test simulations.

The axial stress-strain relationships of the solid constituents in cartilage zones were presented in Fig. 3.10. These zones include the superficial zone (SZ), middle zone (MZ), and deep zone (DZ). At the maximum loading point, the fibrillar component exhibited a stress response change, up to 44% when MO was incorporated, increasing from 0.42 to 0.71 KPa in the MZ. In the other zones, these changes were 9% and 12% in the SZ and DZ, respectively. Regarding the other solid components, the non-fibrillar part showed the highest stress variation, approximately 16%. Thus, the application of MO resulted in notably different stress calculations compared to traditional backward algorithms, highlighting MO's impact on the mechanics of the tissue and further supporting our hypothesis.

Moreover, the stresses recorded in the constituents in the compression experiment for both models showed similar patterns, consistent with previous findings: the osmotic pressure was a major contributor to load resistance, paralleling earlier research by Quiroga et al. 2017; the fibrillar components initially exhibited high stress values, then stabilized to negligible levels, akin to observations in the Wilson et al. 2007 study; notably, the stress in the fibrillar network in DZ was significantly higher than the values in other zones, explained with the higher osmotic pressure and the predominant axial orientation of primary fibrils, enhancing their contribution to axial load resistance.

Figure 3.11 presents the corresponding total stress-strain curves, where both PSAs

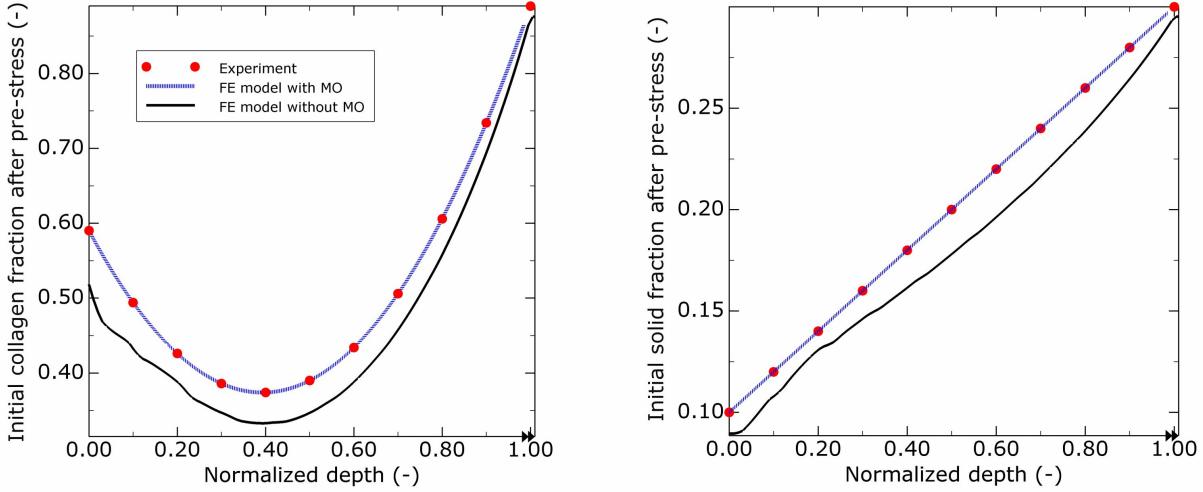


Figure 3.9: Comparative plots of volume fractions with respect to the cartilage normalized depth in the pre-stressed initial state compared with the experimental values [Wilson et al. 2007; Rieppo et al. 2004; Shapiro et al. 2001; Lipshitz et al. 1975]. Abbreviations: MO = material optimization; FE = finite element.

yielded comparable patterns across different cartilage zones. However, minor but noticeable differences were observed, particularly during the relaxation phase, where FE volume changes influenced volume fraction errors. The MO effect was more obvious in the DZ, especially during relaxation, showing about a 12% variation in strain magnitudes. These variations in the SZ and MZ were under 9%. These particular results demonstrate the comparable accuracy of both methods, although the differences could be significant in HF models where stress-strain data correlation is critical [Eskelinen et al. 2019; D. Liu et al. 2020; Stender et al. 2017; Hosseini et al. 2014].

**Joint-scale pre-stressing test.** As depicted in Fig. 3.12, the displacement magnitudes in the post-pre-stressing equilibrium are determined, revealing that the SZ undergoes the most substantial changes (reaching up to the contact points). This phenomenon occurs as this particular zone is neither entirely restricted nor subjected to any external pressure. Consequently, in numerical simulations emphasizing the stress levels in these external zones, the pre-stressing effects might be reasonably omitted. In contrast, the behavior of the deeper layers diverges from this trend.

The deeper AC layers show minimal deformation, attributed to their connection with the subchondral bones. As shown in Fig. 3.13, the deeper layer experiences a fibrillar Mises stress that approximates the overall stress, with values less than 0.2. These findings are in alignment with stresses reported in existing studies for 10% strain in compression tests [Sajjadinia et al. 2019], underscoring the major role of pre-stressing with MO. Moreover, these stress values can be indirectly validated by considering the pre-stresses obtained from the linear triphasic models [Lai et al. 1991; Setton et al. 1996], where stress levels

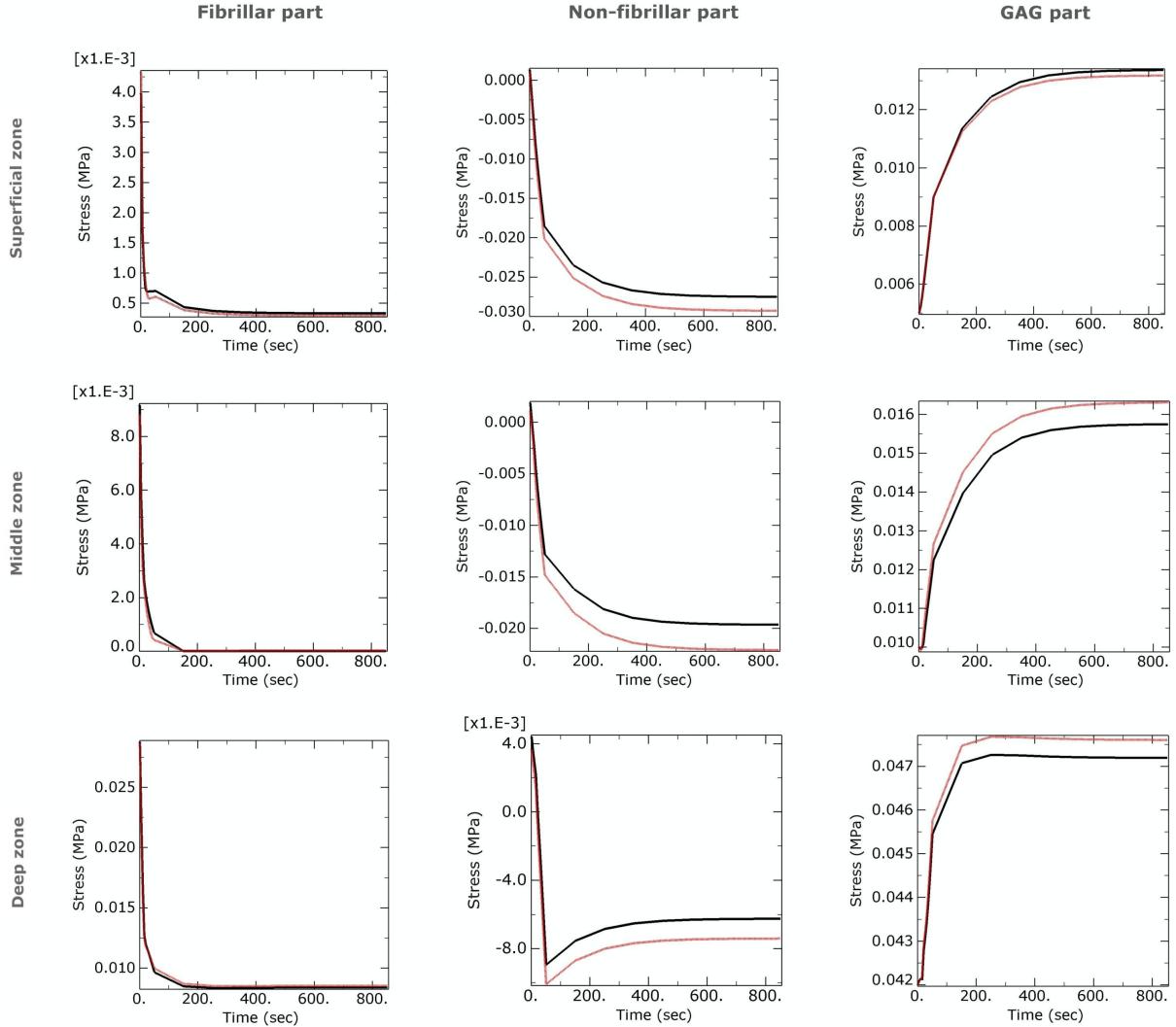


Figure 3.10: Stress vs time plots from numerical simulations of unconfined compression experiments with (black curves) and without (red curves) material optimization for different cartilage zones and effective solid parts after equilibrium.

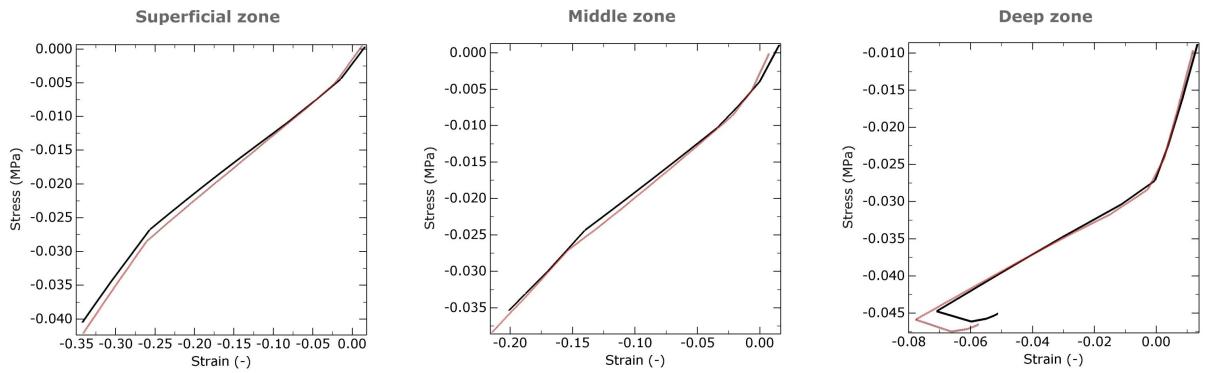


Figure 3.11: Total stress vs strain plots from numerical simulations of unconfined compression experiments with (black curves) and without (red curves) material optimization across different cartilage zones.

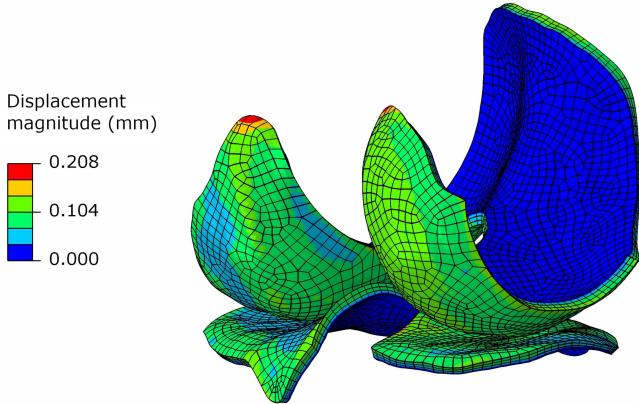


Figure 3.12: Contour plot of pre-stressed cartilage displacements. Note that the displacement values on the cartilage surface accumulate due to the deformation in the underlying layers, resulting in large absolute displacement values. However, this does not necessarily indicate correspondingly large local deformations. These relatively large values are comparable to the regular displacements in the knee cartilage tissue [Chan et al. 2009].

fluctuate below 0.4 MPa based on the direction of the applied load. Compared to them, our large-scale results are also considered reasonable, in light of the different modeling assumptions.

**Other discussion.** This study acknowledges certain limitations. First, the calibration of the numerical model was performed using a specific type of tensile experiment, similar to the foundational calibration test in the related PSA work [X. Wang et al. 2018]. In this way, we approximated the influence of our algorithm on the main constitutive parameters. Second, our PSA might introduce additional complexity into the FPBBS, necessitating the retrieval of deformation gradient values at each integration point. Nonetheless, this is not a significant limitation, as accessing such data is necessary for the FE implementation of our multiphasic model. Notably, our PSA method is comparatively less demanding than the other PSAs that involve element-level manipulations of the FE code, for instance, [Pierce et al. 2015; Weisbecker et al. 2014].

Beyond the implementation efficiency, the convergence of the PSA algorithm was deemed sufficient, as the more time-consuming 3D model implementation took approximately six hours on standard hardware. The runtime will be further reduced in subsequent chapters by applying first-order elements. It is worth mentioning that the application of a unified optimizer significantly accelerated convergence; previously, we tried separate optimizers, which had failed to converge even after half a day using first-order elements.

In summary, the PSA introduced in this study provides a novel and efficient computational approach to be applied to fibril-reinforced and heterogeneous AC models. Our method takes into account the experimentally-observed initial state of both constitutive and geometrical parameters. Several comparative analyses shed light on the importance

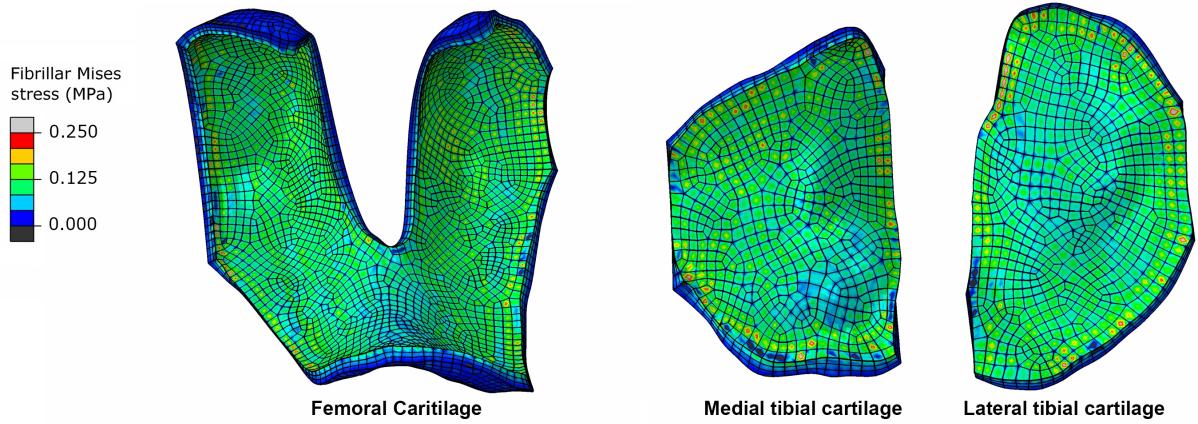


Figure 3.13: Contour plots of stresses in the 3D pre-stressed state. Since the superficial-zone values are negligible, they are not shown here. It is important to note that the constitutive equation for the fibrils is based on the highly discrete orientation of the fibrillar bundles (eq. 2.23), leading to non-physiological concentrations of stress in the model. To mitigate this issue, the selection of higher-order elements or the adoption of a denser mesh could have been beneficial. However, given the good agreement of the results with the previous findings, the use of more computationally expensive meshes was deemed unnecessary.

of such a PSA in the stress evaluation of pre-stressed AC components. Our work also delves into large-scale pre-stressing, showing that pre-stressing can lead to considerable deformation and initial stress distribution. This highlights the criticality of accurately determining the stress-free state. Therefore, this efficient PSA, to be used in the next chapter, is deemed highly relevant in multiphasic AC research, especially in scenarios where osmotic pressure is a pivotal consideration.

# Chapter 4

## Efficient Surrogate Modeling with Hybrid Multi-Fidelity Learning

The iterative nature of numerical modeling in cartilage biomechanics, whether with or without additional complexities such as PSA, renders most of these methods inefficient. As explained in Chapter 2, one solution is to use ML-based surrogates. This chapter introduces a novel surrogate modeling technique using an HML method that combines reduced-order numerical analysis with traditional supervised ML. This method aims to train surrogates of soft tissues like cartilage with a limited number of training samples, as generating HF numerical data with multi-physics models can be very time-consuming. The feasibility of this technique is empirically evaluated and compared with purely data-driven baselines (including GNNs and FFNNs) across diverse scales.

### 4.1 Related Work

**Data-driven surrogates.** The application of surrogate models proved to be an efficient replacement for computationally demanding numerical models [L. Cai et al. 2021; G. Liu 2019; Martínez-Martínez et al. 2017]. Surrogate modeling finds extensive use in biomechanical studies. Examples include deformation analysis of breast and liver employing tree-based algorithms [Martin-Guerrero et al. 2016], prostate modeling using FFNNs [Jahya et al. 2013], and many more. In cartilage modeling, FFNNs have been trained using data from simplified numerical models, e.g., the surrogate of the elastic model in [Paiva et al. 2012]. Another study incorporated biphasic formulations into a small-scale cartilage model surrogate but neglected the tissue’s swelling [Arbabi et al. 2016a]. Further advancements were made by employing multi-physics models in surrogate training, albeit with a significant sample size of 10 000 [Arbabi et al. 2016b]. However, despite their potential, surrogate models are not widely utilized due to the requirement of generating costly numerical samples [Frank et al. 2020].

Utilizing optimized computational code, e.g., through graphics processing unit (GPU) acceleration, can mitigate the above-mentioned issue [Johnsen et al. 2015], although it might demand specific software and hardware not always available [Marinkovic et al. 2019]. Another remedy is to create the surrogate with minimal data under certain conditions, such as reduced simulation scales [Faisal et al. 2023]. Nonetheless, complex multi-physics simulations might still be too costly to be implemented. If the main goal of integrating ML is to increase efficiency, then the models should be trained with as few samples as possible. This is the main reason behind the underutilization of such AI-enhanced modeling.

**Data-efficient learning.** The challenge of small training sets in surrogate modeling can be effectively tackled by integrating physical laws and domain-specific insights into the training methodology. Such integration, as shown by Kashinath et al. 2021; Linka et al. 2021; Hoerig et al. 2020, has been shown to enhance performance even with limited data. These methods, despite their evident success, are still evolving and require meticulous design and complex coding. Often, they introduce new constraints to the training, such as loss function penalization in standard learning algorithms. Challenges particularly arise in dealing with multi-physics scenarios and discontinuities in mechanical models, as discussed in Coutinho et al. 2023; S. Cai et al. 2021; Karniadakis et al. 2021; Fuks et al. 2020.

Another approach to managing limited data involves simplifying the numerical model to focus on key information, thus reducing the number of necessary parameters. This method is exemplified in studies by Pellicer-Valero et al. 2020; Niroomandi et al. 2012. The efficiency gains from such model order reduction (MOR) methods have sparked interest in various domains, e.g., the virtual reality applications in surgical simulations [Cueto et al. 2014]. However, these methods also present challenges in implementation, as noted in the Lauzeral et al. 2019 study. Considering this, multi-fidelity methods in the MOR offer an alternative, potentially implemented with non-destructive methods (without directly affecting the numerical solvers). These techniques have shown promising results across different fields, implemented with diverse ML methods [Ahn et al. 2022; L. Zhang et al. 2022; Cheng et al. 2021; Yang et al. 2019].

A prevalent strategy in multi-fidelity MOR-based surrogate modeling involves using an LF model in conjunction with an ML model to enhance the LF output accuracy, as detailed by Peherstorfer et al. 2018. Such models have found applications in diverse fields using various LF modeling implementations [L. Zhang et al. 2021; Q. Zhou et al. 2017]. However, to our knowledge, no implementation has been done using LF modeling for multi-physics soft tissue models, like cartilage, as proposed in our study. The proposed approach utilizes an LF numerical part that simplifies the multi-physics equations, focusing on the tissue's approximate behavior. The accuracy of the LF outcomes is then enhanced through an ML model to match the HF model's precision with non-destructive HML implementation.

## 4.2 Methodology

### 4.2.1 Efficient Hybrid Learning

In the following of this chapter, we use the term ML to primarily denote the conventional ML-based predictive models, except when specified otherwise, as illustrated in Fig. 4.1(a). In a standard supervised learning approach, an FE model generates various informative training samples at distinct analysis points using a comprehensive physical model. These HF outcomes serve as the target ML dataset, while the input features may include various physical parameters, such as boundary conditions and material properties.

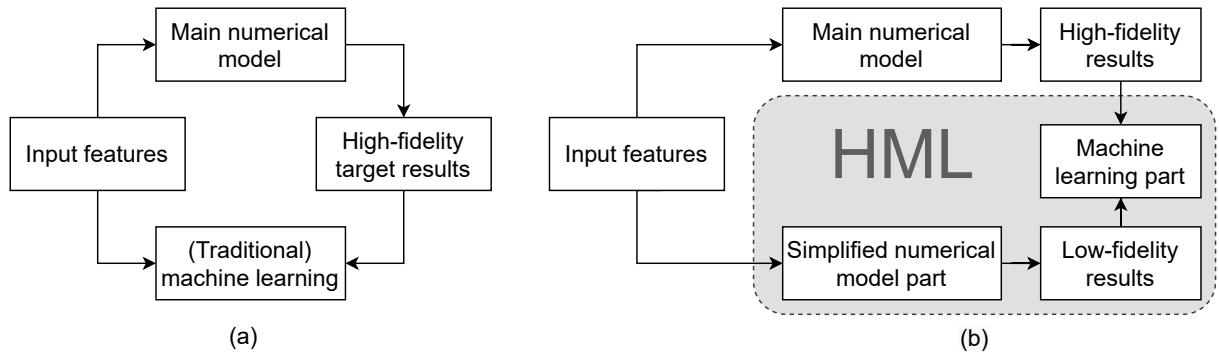


Figure 4.1: Illustration of a standard machine learning surrogate model (a) and its integration in a hybrid machine learning (HML) framework, as connected with a simplified finite element model sequentially (b).

In the proposed HML approach, we initiate with an LF model derived by simplifying the HF equations. This computationally efficient model provides a basic estimate of the system's behavior and precedes the ML component, which aims to refine these initial predictions to match the accuracy of the HF model. In essence, the ML component's input features are derived from the outputs of the LF model, as depicted in Fig. 4.1(b).

While this methodology offers considerable flexibility in choosing the LF model, it necessitates the creation of two distinct training datasets using varied computational techniques. Nevertheless, this may not be a significant limitation, as the LF model is just the simplified and more efficient version of the main HF model. Our empirical analysis, explained in the next section, indicates that this approach may also require fewer training samples overall to achieve comparable accuracy and potentially offers faster training convergence; thereby considerably mitigating the high computational demands typically associated with training and data generation.

For instance, a complex HF physical model might be defined by using biphasic equations. In contrast, a corresponding LF model is the same FE model but could be represented by simpler monophasic equations, which, while not capturing the intricacies of each phase, approximates the overall system response. Depending on the specific physical

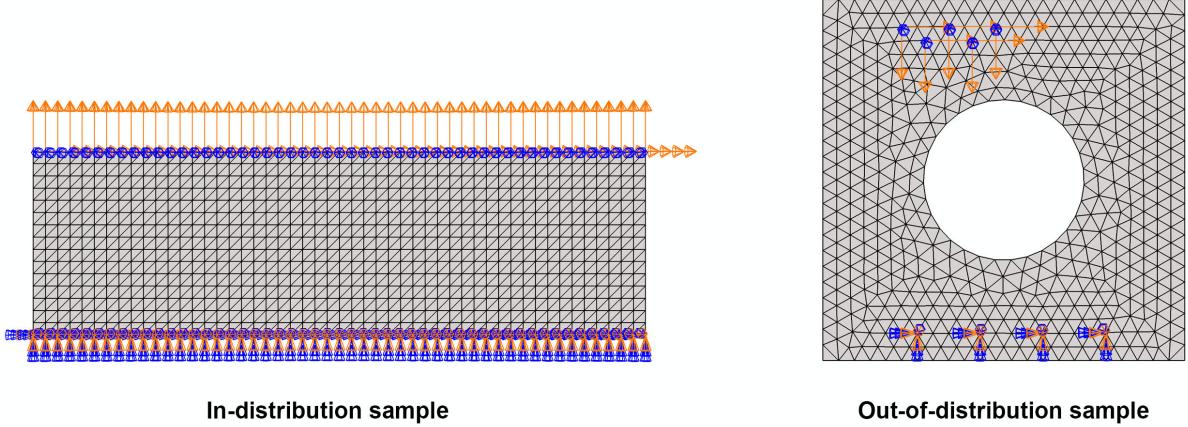


Figure 4.2: Illustration of various small-scale computational samples for surrogate modeling.

problem, various combinations of equations can be employed. Accordingly, in the following, we explore two different selections of LF modeling to test the applicability of this method.

#### 4.2.2 Simulation Tests

We develop two computational physics problems for generating datasets, via FE modeling. These problems replicate standard two-dimensional (2D) and 3D tests on artificial soft tissues, which are commonly employed in repetitive biomechanical simulations. The mathematical formulations and their computational implementations are elaborated in Chapters 2 and 3.

**2D problem.** This involves a small-scale problem where a soft biological material is subjected to 2D load testing. This is depicted in Fig. 4.2 (left). We model a rectangular specimen of dimensions  $0.3 \times 1.0 \text{ mm}^2$  using a hybrid FE model. A porous hyperelastic material model is used (as per eq. 2.24 with  $\emptyset_0^S = 0.15$ ), along with osmotic pre-stress (see eq. 2.17 with  $\alpha_2 = 3.22$ ). Here,  $\alpha_1$  represents the adjustable input parameter in our surrogate model. We deliberately omit fibrillar and fluid constituents for computational efficiency. Obviously, our PSA is essential for initializing the numerical experiments, while the LF model is simplified by excluding the pre-stressing feature.

To assess the model’s capacity to handle out-of-distribution scenarios, we also evaluate a different 2D setup; see Fig. 4.2 (right). This setup involves different load applications and geometry, specifically a  $0.9 \times 0.9 \text{ mm}^2$  specimen with a central circular hole of  $0.2 \text{ mm}$  in radius. In both simulations, we fix the bottom surface of the tissue and apply displacement conditions at the top. For the mesh, the linear and plane strain triangular elements are used.

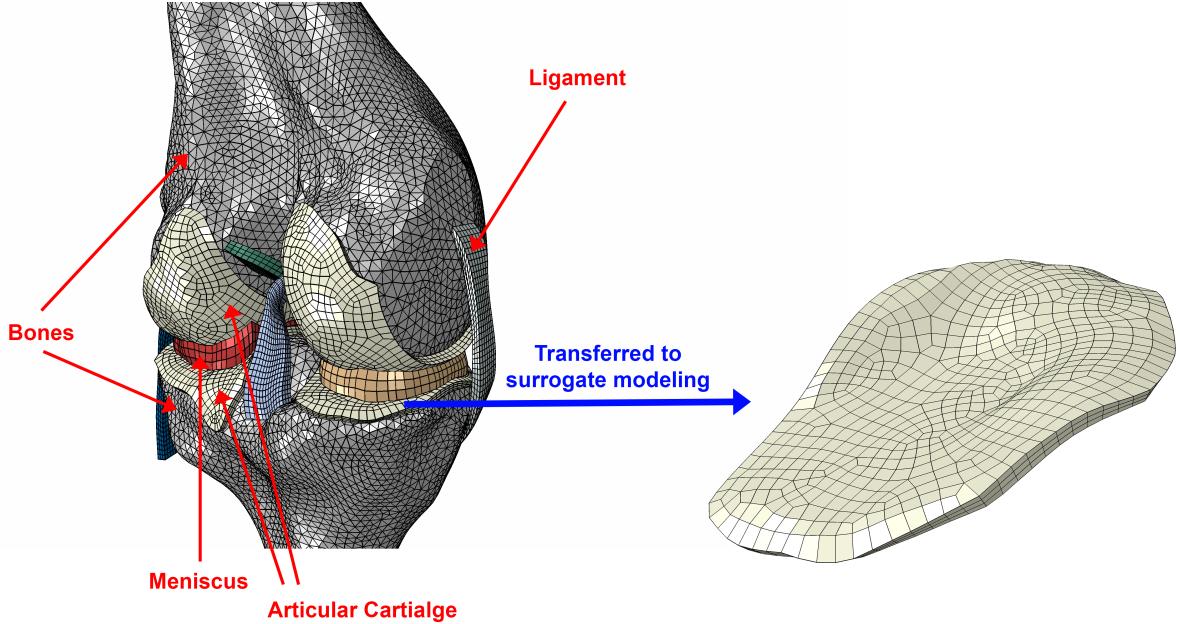


Figure 4.3: Numerical mesh of the large-scale problem used for creating surrogate models.

**3D problem.** Our large-scale simulations apply the tibiofemoral model introduced in Chapter 3, but they utilize different multi-physics equations. This test aims to analyze the influence of varying body weights on one of the cartilage substructures (refer to Fig 4.3). We implement a hyperelastic equation for LF modeling, based on eq. 2.13, where  $C_{10} = 2.48 \text{ MPa}$  and  $D_1 = 0.59 \text{ MPa}^{-1}$ , whereas the HF model also adds an inviscid water component to the constitutive equations.

In the biphasic model for HF simulations, we use first-order elements to accelerate data generation. The permeability parameters are derived from existing literature on the cartilage MZ [Sajjadi et al. 2019; Stender et al. 2017]. While the fluid is allowed to move freely across the cartilage surfaces, the weight is applied uniformly across the femur, with the fixed tibia and no flexion-extension motion.

**Datasets and evaluation.** For the datasets of the 2D models, 20 representative samples are generated. These samples are derived from a uniform distribution ( $U$ ), encompassing two displacement-based boundary conditions within  $U(-0.1 \text{ mm}, 0.1 \text{ mm})$ , a constitutive variable  $\alpha_1$  ranging from  $U(0.005, 0.010 \text{ MPa})$ , and a one-hot vector to distinguish the fixed nodes with the others. Minor noise is introduced at each node to enhance model generalizability. Thus, the nodal coordinates are additional input features, while the introduced noise impacts the initial conditions and influences all subsequent model outputs.

The outputs from both LF and HF simulations consist of computed nodal displacement and total stress values. For model validation, we generate 10 additional in-distribution samples with varied inputs. Our test set, which also comprises 10 samples, employs the

out-of-distribution FE model, testing the model’s adaptability to unseen conditions.

For the large-scale surrogate model, we create 100 samples. Inputs for the HF model include a range of the weight from  $U(-900 \text{ N}, -400 \text{ N})$ . Outputs cover determined normal stress and strain at each node, whereas the LF outputs focus on nodal deformation. We randomly distribute these samples across training, validation, and testing subsets, with the validation set sized at a third of the training set. The exact count of training samples is noted in the results, emphasizing the scarcity of such data.

Notably, the range of 2D problem results generally falls under a similar range between  $10^{-1}$  and  $10^{-3}$ , considering their units and absolute values. Thus, we employ an unscaled MSE metric for all node outputs in all the evaluation samples. This allows simple comparison between different training conditions. Conversely, for the 3D models, due to the vast difference in input-output scales, target feature normalization based on training samples is necessary. Although this normalization facilitates comparison across different outputs, the use of MSEs can sometimes be deceptive, as it may average out anomalous nodes with scaled outputs. To address this, we use the PMSE metric, averaging these scaled values individually for each node.

**Training settings.** Our approach adapts the previously used MeshGraphNets method [Pfaff et al. 2021] for our specific 2D surrogate model needs. The original framework, based on a GNN, incorporates a series of FFNNs organized into encoder, processor, and decoder modules [Sanchez-Gonzalez et al. 2020; Battaglia et al. 2018]. This architecture facilitates local learning at each node and connected edges, enhancing spatial equivariance through bidirectional edge sets. The core model initially encodes node and edge features into latent representations, which are then sequentially processed by message-passing blocks. While the residual connections further augment the latent vectors, enhancing the model’s convergence [Sanchez-Gonzalez et al. 2020], the final latent representation is decoded into the surrogate outputs.

We diverged from the standard MeshGraphNets implementation by omitting domain-specific configurations, particularly the separate edge types, as they were irrelevant for our 2D tests lacking mechanical contact. Also, instead of the original noise strategy, we use our aforementioned noise generation method. We also revised the preprocessors and evaluation algorithms of the trained model, tailoring them for static system simulations of this study that focus solely on end conditions as outputs.

To develop 2D model 1, we employ a training method that accommodates various input sample sizes, utilizing an  $L_2$  loss function for nodal results. The training involves a maximum of 1 000 steps, with a learning rate of  $10^{-3}$  and the first (100) steps dedicated to initializing the online normalizer [Sanchez-Gonzalez et al. 2020]. The model’s FFNNs components comprise three hidden layers, and we included three message-passing blocks (containing such FFNNs) with a latent size of 40. This size is also the same as the encoder

output size and the decoder input size.

Furthermore, the edge sets are constructed from relative nodal distances and their euclidean norms, with the encoder’s input shape set to size 3. In the hybrid model, the node set is made by the nodal outputs, leading to an encoder’s input size of 6 and a decoder’s output size of 6. The batch size is set to the total node count of each sample (i.e., 816), and the rest of the hyperparameters are identical to the original study.

Our study also includes an empirical comparison of 3D models built using standard FFNNs. We optimized these models for rapid convergence by maintaining a minimal neuron and layer count. After extensive manual tuning of hyperparameters, 3D model 1 was configured with four hidden layers, each containing 940 neurons equipped with the ReLU activation functions and subsequent normalization layers. The output dimension matches the size of the total nodal results of each one of the HF samples (i.e., 94 032). The input dimension of the ML surrogate is 1, while for the HML model’s ML component, it equals the total nodal results of each of the LF samples (i.e., 8 364). Training is conducted on a limited number of samples with a  $10^{-4}$  learning rate. For the other settings, we adhere to the default library settings.

While we also open-sourced<sup>1</sup> all the code for further details, in the following section, we will detail additional models, their configuration, and training to assess the impact of hyperparameters and training sample sizes.

### 4.3 Results and Discussion

Empirical tests were conducted using various LF and HF computational models on a regular computer. The efficiency gains are illustrated in Fig. 4.4 using a violin plot [Hintze et al. 1998]. These gains underscore the practicality of employing LF models over HF ones. Variations in the loading rate applied in the 3D simulations led to a broader range of computation times. It is important to note that in real-world scenarios, this disparity might be more pronounced, given that the HF simulations were modified in this study to expedite the process of surrogate model development.

In our analysis, the runtime of ML inference was not considered as it occurred in real time, and the majority of computational expenses were attributed to the generation of numerical data. In the following, we provide a comparative analysis of HML and ML methods across different simulation scenarios, highlighting the efficacy of HML models when using a limited number of HF samples. Conducting a couple of independent multi-physics tests with varying surrogate modeling assumptions enabled an effective assessment of model performance under diverse training settings.

---

<sup>1</sup><https://github.com/shayansss/hml>.

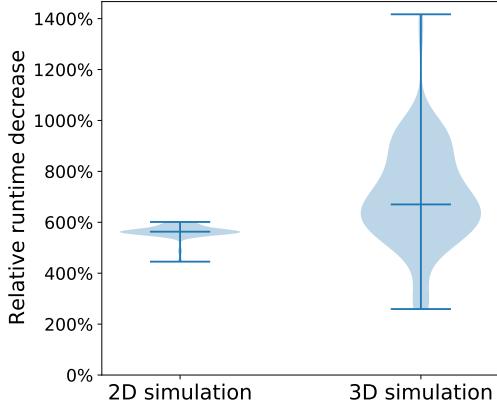


Figure 4.4: Representation of the relative decrease in runtime for low-fidelity simulations compared to high-fidelity simulations. This data is obtained by calculating the ratio of the runtime reductions to the runtimes of the low-fidelity simulations.

**2D simulation.** The evaluation of MSEs for both in- and out-of-distribution sets, in the context of 2D simulations, is depicted in Fig. 4.5. We deliberately use small datasets, as it is usually the case for soft material surrogate modeling [G. Liu 2019]. The HML frameworks demonstrate superior performance compared to ML models. The errors in most cases for HML models are at or below  $10^{-4}$ , with a notable exception being the error for the hybrid surrogate trained with four samples. However, this error is still lower than any error observed in the ML models. Further experiments were conducted using a different setting (model 2), characterized by less efficient, yet increased learnable parameters, as it incorporates three additional message-passing phases as the principal hyperparameter [Pfaff et al. 2021]. Even with a learning rate set to  $10^{-4}$  and training steps increased by a factor of eight, the results remained consistent, underscoring the precision and generalizability of the HML models.

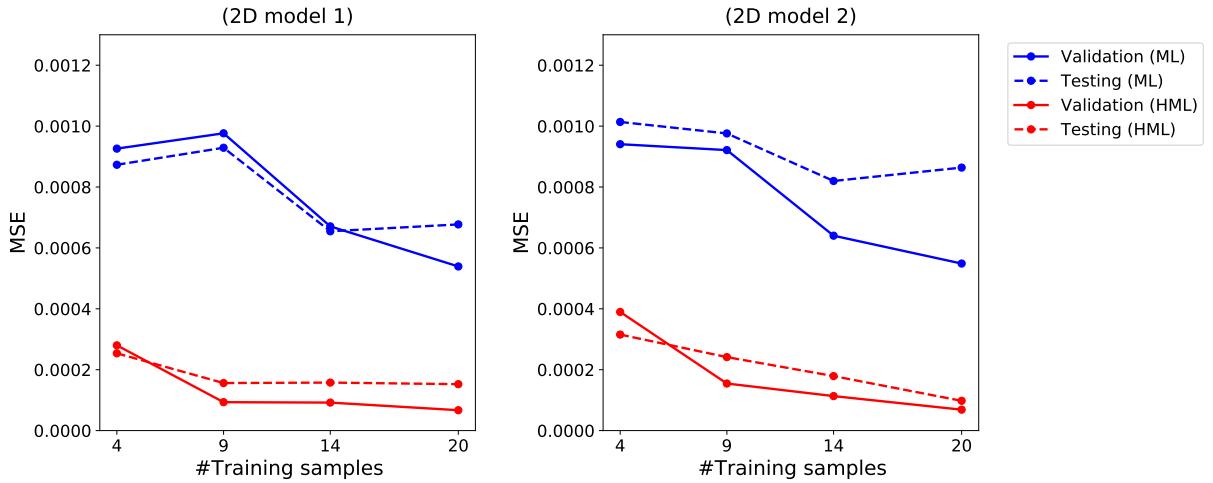


Figure 4.5: Assessment of trained models with different numbers of training samples in 2D tests, implemented via machine learning (ML) and hybrid ML (HML).

Furthermore, the HML model’s methodology, mapping input features into a new representation for subsequent data-driven analysis, bears resemblance to feature-based transfer learning. In such transfer learning, source features are transformed to facilitate knowledge transfer from one domain to another [Zhuang et al. 2021]. Unlike common data-driven models, which employ feature transformation strategies [S.J. Pan et al. 2011; Dai et al. 2007; Daumé III 2007], the HML model uses a physics-based algorithm, the LF model, for transforming input data into more informative representations. This approach mirrors the transfer learning concept and elucidates the enhanced performance of the HML model under the constraints of efficient training settings and limited training data availability.

An additional aspect of our study involved assessing how the accuracy of LF modeling impacts the performance of HML modeling. For this, we replicated the 2D tests using the same HML, but substituting LF results with HF outcomes. As presented in Table 4.1, this modification yielded errors comparable to the original LF inputs, indicating that the model, already adeptly trained with LF data, managed to perform effectively despite the inherent stochasticity of the training process. This observation suggests that even potential inaccuracies in LF modeling, as also reported in Chapter 3 and other PSA studies [Bols et al. 2013; X. Wang et al. 2018], do not hinder its ability to generate an informative representation for the downstream ML part.

Table 4.1: Recorded mean-squared errors ( $\times 10^{-3}$ ) on validation/testing sets in 2D modeling, implemented via machine learning (ML) and hybrid ML (HML).

Surrogate model	Training set size			
	4	9	14	20
ML 1	0.93 / 0.87	0.98 / 0.93	0.67 / 0.65	0.54 / 0.68
HML 1	0.28 / 0.25	0.09 / 0.16	0.09 / 0.16	0.07 / 0.15
HML 1 (with high-fidelity inputs)	0.33 / 0.26	0.10 / 0.16	0.07 / 0.13	0.06 / 0.12
ML 2	0.94 / 1.01	0.92 / 0.98	0.64 / 0.82	0.55 / 0.86
HML 2	0.39 / 0.32	0.15 / 0.24	0.11 / 0.18	0.07 / 0.10
HML 2 (with high-fidelity inputs)	0.32 / 0.35	0.09 / 0.16	0.16 / 0.17	0.06 / 0.12

**3D simulation.** As depicted in Fig 4.6, the  $l_2$  loss values captured during the training phase are compared. The 3D model 1, incorporating the HML framework, demonstrated superior performance over its ML counterpart. Another variant, 3D model 2, adhered to an identical sample regimen (comprising three training and one validation sample) yet employed the ELU activation function. The outcomes of this model variation were not markedly different, with the HML surrogate still having a negligible error. A critical observation here is the rapid and efficient training process, as it only used 100 epochs.

Further exploration into the hybrid methodology was conducted through 3D model

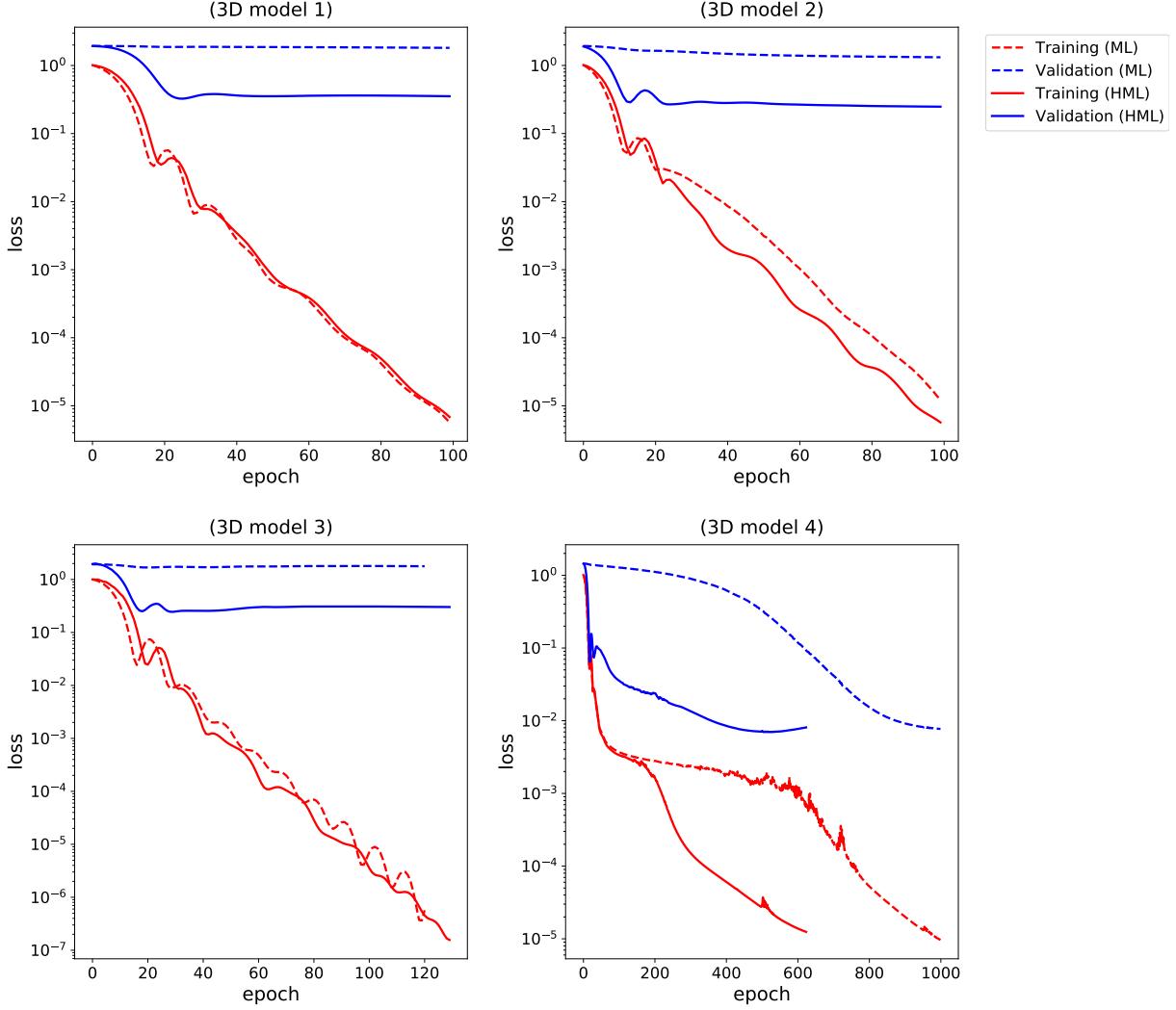


Figure 4.6: Comparative analysis of loss values against the number of epochs in 3D surrogate modeling. It is observed that certain models achieved convergence earlier owing to the existence of the early stopping mechanism. Abbreviations: ML = machine learning; HML = hybrid machine learning.

3 (as also summarized in Table 4.1 for the loss values in the final epoch). This model replicated the training and validation sample sizes of its predecessors, but introduced significant modifications in the epochs and neurons count - a tenfold increase and decrease in the former and latter, respectively. Additionally, the ReLU function was utilized alongside an early stopping parameter set at 100. Despite these alterations, the efficiency of training was maintained, even over an extensive number of epochs. However, it is noteworthy that these adjustments harmed the HML performance, as evidenced by a surge in validation loss after approximately 30 epochs, yet it still outperformed the ML surrogate.

For 3D model 4, the size of training and validation datasets was increased by a factor of three. We maintained hyperparameters akin to those in our first 3D model but extended the training to a maximum of 1,000 epochs, and we used the early stopping

Table 4.2: Loss values at the final epoch on training and validation sets in 3D modeling, implemented via machine learning (ML) and hybrid ML (HML).

Surrogate model	3D model 1	3D model 2	3D model 3	3D model 4
ML (Training)	0.0000057	0.0000123	0.0000006	0.0000096
HML (Training)	0.0000068	0.0000057	0.0000002	0.0000125
ML (Validation)	1.8169936	1.3181295	1.7773948	0.0076983
HML (Validation)	0.3536258	0.2480008	0.3019600	0.0080819

criteria established in our third 3D model. This escalated the computational cost of our last 3D model significantly. Indeed, the increased data generation, more learnable parameters, and extended training duration led to a substantial increase in both training and computational time for numerical data generation. The findings indicate that both the ML and HML models yielded comparably consistent loss values upon convergence. Notably, the HML approach demonstrated a markedly faster convergence rate, recording a validation loss of 0.1 within approximately 10 epochs – approximately 75 times quicker than the ML model.

To further interpret the performance disparity observed between the models, Fig. 4.7 presents a comparative analysis of the PMSE contours, applied to a reference state of one cartilage substructure. The first three ML models displayed suboptimal performance across various points, whereas their HML counterparts achieved significantly higher accuracy levels, with minimal pointwise error outliers (evidenced by sparse small red areas). The fourth ML model exhibited comparable accuracy levels to the HML model, with minor differences. However, in this inefficient surrogate modeling scenario, there was also a greater frequency of pointwise errors, suggesting a preference for the other more efficient HML models. Again, this analysis emphasizes the benefits of adopting HML modeling techniques, particularly in scenarios with limited data availability.

**Other discussion.** This work acknowledges certain limitations. Firstly, we simplified the HF numerical models for efficient data acquisition; however, this does not majorly impair the study as the simulations primarily aim to highlight the efficacy of the hybrid approach in biomechanical simulations in general, rather than serving as clinically validated AC models. Secondly, the empirical experiments presupposed that the training phases should not substantially exceed the numerical data generation in cost, to justify the adoption of these surrogate models. Consequently, expensive settings for surrogate training were not employed. While this may restrict our findings to the specific conditions of our models, such presumptions are often necessary when utilizing surrogate models on regular hardware settings.

In summary, this chapter introduces an HML methodology for multi-fidelity-based surrogate modeling and FE analysis of cartilage-like soft tissues. The LF model in this

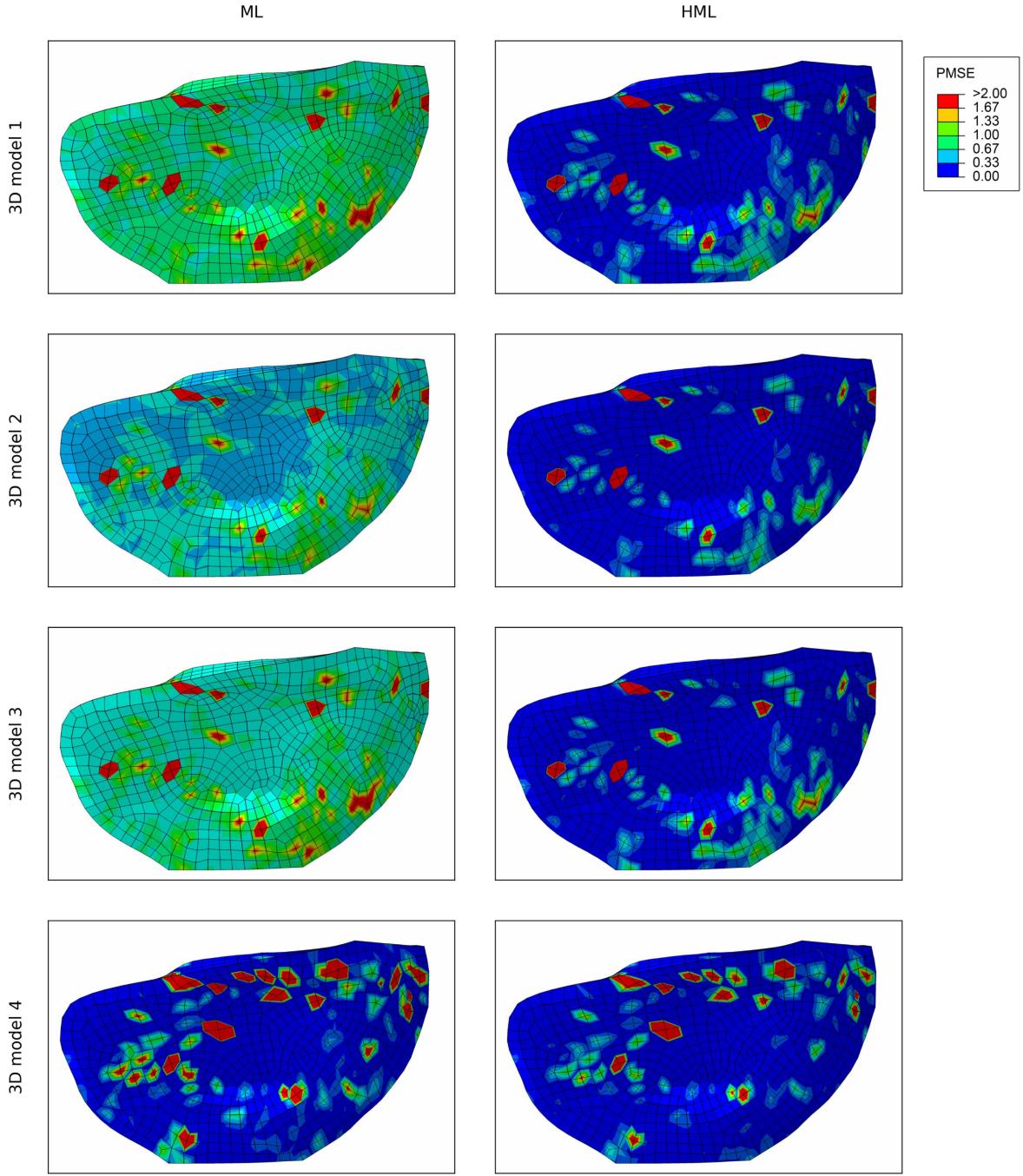


Figure 4.7: Illustration of pointwise mean squared error (PMSE) contours for all three-dimensional surrogate models. Compared to regular machine learning (ML), the hybrid ML (HML) achieves satisfactory accuracy in most areas; however, high local errors may persist at a few nodes. These errors are likely due to numerical modeling inaccuracies or the compromises made during manual hyperparameter tuning. Specifically, hyperparameters were chosen based on their ability to rapidly converge and for their capability to reduce the order of the latent space efficiently, allowing models to complete training within a few minutes. While this approach helps maintain focus on general model performance, it might overlook some local complexities. With higher-quality data and a more intensive hyperparameter search, these local errors could potentially be minimized.

context is derived by simplifying the multi-physics formulations, making the MOR process relatively direct and non-invasive. Extensive empirical evaluations also reveal that our hybrid approach enhances training efficiency and improves the performance of surrogate modeling with minimal training data. This is particularly beneficial, as it overcomes the challenge faced by many existing ML-based surrogate models that usually require numerous costly HF samples.

# Chapter 5

## Efficient and Generalizable Learning for Surrogate Modeling

The effectiveness of our HML in single-scale, few-shot learning has been demonstrated, with the GNN model showing promise for generalizable learning. Our goal is to develop a multiscale surrogate model, trained on augmented tissue-scale data, capable of potentially achieving zero-shot generalizability for knee-scale simulations. This chapter introduces a novel, physically constrained perturbation method to DA, generating various elements, that might be seen in large-scale analyses using small-scale data. We also extend our hybrid algorithm with several preprocessing and customization algorithms, including the integration of an upstream AE model to refine input features, and the application of feature transformation, resampling, and cost-sensitive functions to address data imbalances. This is complemented by a customized, memory-efficient framework that accelerates training. We test this approach using the advanced multiphasic model of cartilage, considering all its constituents, to evaluate the viability of this technique and compare it to its ablated versions (wherein certain extensions are disabled for the sake of comparison).

### 5.1 Related Work

**Generalizable surrogate modeling.** The challenge of generalization in ML-based surrogate models, especially for models involving different topologies, has been explored in few studies [Hoffer et al. 2022; Taghizadeh et al. 2020]. A prominent approach to address this issue involves the integration of theory-inspired features or physics-constrained models. In particular, H. Gao et al. 2021 successfully applied a theoretically-inspired feature engineering method to adapt their model from a regular to an irregular mesh. Also, Nourbakhsh et al. 2018 developed a surrogate model for truss analysis utilizing domain-specific features to encapsulate various topologies. The complexity of integrating domain knowledge, especially in cartilage biomechanics, as pointed out in the previous chapter,

favors the use of graph neural networks for their capability in generalizable modeling across different scales. This data-driven approach can employ the relational inductive biases [Battaglia et al. 2018], as successfully implemented in the single-scale 2D simulation of the previous chapter. Overall, such methods demonstrate potential, as facilitated multiscale generalization through their message-passing and local aggregation functions [Battaglia et al. 2018; Sanchez-Gonzalez et al. 2020].

Nonetheless, a primary challenge in training graph neural networks lies in their requirement for extensive message-passing steps, which can lead to prolonged training periods and the oversmoothing issue in downstream layers, where they become mostly identical and useless [X. Zhang et al. 2024]. A solution lies in limiting the message-passing iterations by adopting multi-level mesh resolutions, as extensively researched [Cao et al. 2023; Fortunato et al. 2022; Z. Li et al. 2020]. Our hypothesis posits that the HML technique could mitigate this challenge without relying on such complex input meshes. Additionally, we integrate an upstream AE in our custom memory-efficient training loop, a widely adopted technique in surrogate modeling for tasks like reduced-order modeling or feature denoising [Fresca et al. 2022; Pant et al. 2021; Fresca et al. 2021], to possibly accelerate the training of our multiscale cartilage surrogate model.

**Augmented graph data.** DA, the process of synthetically generating new data, is critical when dealing with limited datasets. Common augmentation methods include the application of generative models, noise addition, and the application of problem-related transformations such as rotations to the image data [Mumuni et al. 2022]. Such techniques have shown notable efficacy in enhancing model performance, particularly in physics applications where configuration-dependent tensors are utilized [Cheung et al. 2023]. Alternatively, the Mixup technique, involving the convex combination of sample pairs to interpolate new data, is another widely adopted method across various domains, including imaging [Guo et al. 2019; H. Zhang et al. 2018], natural language processing [Jindal et al. 2020; Guo 2020], and graph data [Han et al. 2022; Y. Wang et al. 2021]. However, it has been mostly applied to graph classification rather than regression problems like surrogate modeling, an area where our technique offers a novel solution.

In the physics context, innovative, problem-specific augmentation techniques have been developed. More specifically, X. Zhao et al. 2021 introduced a method in temperature prediction that exponentially increases training data by applying pairwise differences instead of absolute values. Parida et al. 2023 also augmented earthquake signals using a discrete wavelet transform method, creating new samples by perturbing the wavelet coefficients of original signals. Additionally, many surrogate models trained on image-based snapshots, e.g., [Dong et al. 2022]; thus employed random rotations to ensure orientation invariance, and in some multi-fidelity studies, e.g., [Sella et al. 2023], LF training samples directly augment datasets. Our research particularly augments datasets for multiscale

surrogate modeling without any large-scale data, a novel approach in this context.

**Handling imbalanced data.** Given the skewed distribution prevalent in biomedical data [Rauschenberger et al. 2020], we hypothesized that our multiscale cartilage data necessitates specialized treatment to avoid uninformative representations biased towards common, simpler nodal representations. Traditional methods, such as applying logarithmic or cubic transformations based on the data distribution [Bhattacharya et al. 2017], help stabilize variance and reduce skewness. The Yeo–Johnson transformation [Yeo et al. 2000] represents a combined technique, defined by:

$$\phi(x; \lambda) = \begin{cases} [(x + 1)^\lambda - 1]/\lambda & \text{if } \lambda \neq 0, x \geq 0, \\ \ln(x + 1) & \text{if } \lambda = 0, x \geq 0, \\ -[(-x + 1)^{2-\lambda} - 1]/(2 - \lambda) & \text{if } \lambda \neq 2, x < 0, \\ -\ln(-x + 1) & \text{if } \lambda = 2, x < 0. \end{cases} \quad (5.1)$$

Here,  $\phi$  is the transformed value of  $x$  and  $\lambda$  is a hyperparameter determined through maximum likelihood during data preparation. Other methods include maximal loss focusing on top error percentages and weighted loss, with the weights either statically assigned based on feature frequency [Steininger et al. 2021] or dynamically based on training errors [Fernando et al. 2022]. Resampling graph partitions or subgraphs depending on their importance, either statically by initial nodal feature distribution or dynamically by nodal error distribution during training, offers another solution. Although these methods are predominantly implemented in graph classification tasks [Y. Wang et al. 2022; Zeng et al. 2020; Chiang et al. 2019], we have adapted them for our regression-based surrogate modeling problem.

## 5.2 Methodology

### 5.2.1 Efficient and Generalizable Learning

We initially try to balance and centralize the data to not only enhance training but also to ensure a fair evaluation. This is primarily achieved through the Yeo–Johnson transformation (eq. 5.1), along with other similar transformations that can be evaluated after delving into the data. Additionally, applying cost-sensitive and resampling strategies further complements this by compelling the trainer to focus more on the complex or rarer nodal points. The simplest solution is to utilize the maximal loss [Shalev-Shwartz et al. 2016], which restricts training to the top 10% of the highest nodal errors.

Additionally, in our proposed methodology, each dimension of the dataset, denoted as  $\mathbf{D}$ , is assigned a weight based on its frequency distribution. This process commences with

the discretization of  $\mathbf{D}$  into  $n$  distinct bins. These bins are uniformly spaced according to the range of values present in  $\mathbf{D}$ , defined by its minimum and maximum values. For a particular value  $d \in \mathbf{D}$ , the corresponding bin index is represented by  $\text{bin}(d)$ , and then the frequency  $f_i$  of the  $i^{\text{th}}$  bin is calculated as the inverse of the total count of elements in  $\mathbf{D}$  that are categorized within that bin. Subsequently, these frequencies are normalized to ensure their sum equals 1. This normalization process yields the final weight  $w_i$  for each value of the bin, which assuming we have  $n$  bins, is calculated as:

$$w_i = \frac{f_i}{\sum_{j=1}^n f_j}. \quad (5.2)$$

In this context,  $\mathbf{D}$  can be seen as the original nodal input features, with a higher focus on the less frequent values that persist even after pre-preprocessing. Alternatively,  $\mathbf{D}$  could be dynamically assigned, based on the loss errors at each training epoch, to prioritize more complex nodes during the training phase. The weights thus assigned are then incorporated multiplicatively with the  $l_2$  loss errors to implement a cost-sensitive function during training.

An alternative approach involves the application of subgraphing, i.e., sampling of the graph partitions, as illustrated in Fig. 5.1. Studies, e.g., [You et al. 2020; F. Sun et al. 2020; Veličković et al. 2019], have shown that applying this may further increase consistency between local and global representation. We define a subgraph for each node before training, which is formed by identifying indices of neighboring nodes and edges. This identification is implemented through message-passing steps that accumulate the indices of the nodes and edges. Each subgraph of each node, called the central node, is then isolated based on the collected indices. Considering that graph partitioning can affect each neighboring node, the training errors are only computed on the central nodes, which aggregate all the unaffected information. The selection of these central nodes for subgraphing is guided by the calculated weights, now interpreted as sampling probabilities. The resampling can again be implemented statically and dynamically, as implemented for the cost-sensitive functions.

On the other hand, we introduce a DA technique that perturbs each nodal feature, denoted as  $\mathbf{T}$ . This modification is inspired by the graph-based Mixup method [Guo et al. 2023; Park et al. 2022] and nodal graph perturbation [Gibson et al. 2022] but is adapted to suit the nature of our spatiotemporal data, as follows:

$$\mathbf{T}_{\text{new}} = K\mathbf{T}_i + (1 - K)\mathbf{T}_j. \quad (5.3)$$

Here,  $K \in [0, 1]$  is a random value,  $\mathbf{T}_{\text{new}}$  represents the newly created nodal feature. It is derived from a combination of features corresponding to two existing points,  $\mathbf{T}_i$  and  $\mathbf{T}_j$ , located in the spatial axis and temporal space.

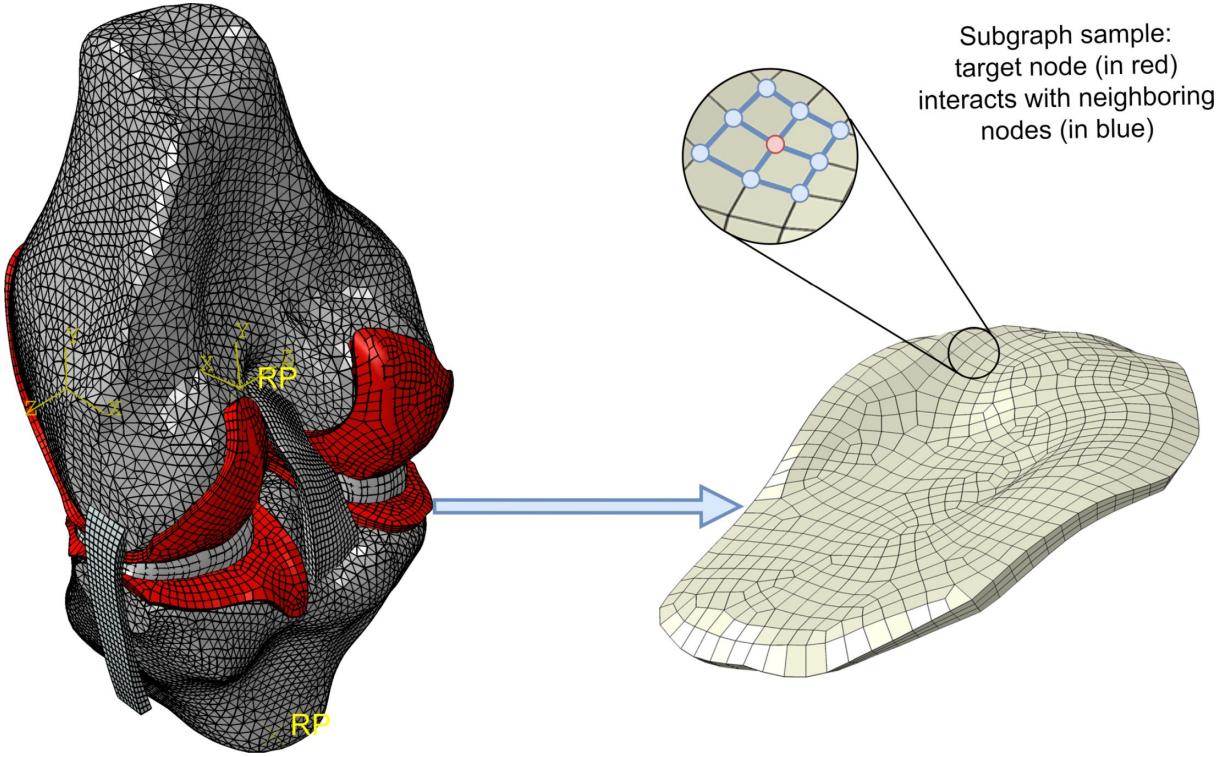


Figure 5.1: Example of a central node of a one-hop subgraph, shown in red, interacting with neighboring nodes, shown in blue, within a numerical cartilage model. By subgraphing, the semantics can be preserved in the randomly selected central (target) nodes.

To implement our DA, we first need to identify and move a selected frame and node of the  $i$ -th point used for linear interpolation by randomly selecting  $K$  at each point and selecting one of the closest points, i.e., the  $j$ -th point. Finding associated temporal frame points is straightforward: we simply select the closest pair of recorded temporal frames, in the temporal axis, and one of these points is then randomly replaced with the new point as long as it is not the initial or final frame points. Such methods are widely used in DA of time series [Oh et al. 2020], which now we implement on each node. However, the same approach does apply to DA of geometry. If we were to perturb nodal points randomly for later augmentation by interpolation, there is a risk that the new nodal points might fall outside the relevant geometry, generating invalid shapes, especially around contacting regions.

Our DA is capable of generating a valid shape, which may be present in large-scale simulations. This process is accomplished by randomly selecting one of the *allowable* connecting edges at each node. Then, as illustrated in Fig. 5.2, the selected edge is used to relocate the node, provided that it does not affect the boundary. Consequently, we define our associated  $i$ -th and  $j$ -th points (in this context, nodes) as the chosen node to be moved (essentially, to be replaced by a new node interpolated using the previously mentioned equation) and the node to which it is connected. This implies that the nodes are confined to moving along the selected allowable edges. Identifying these allowable

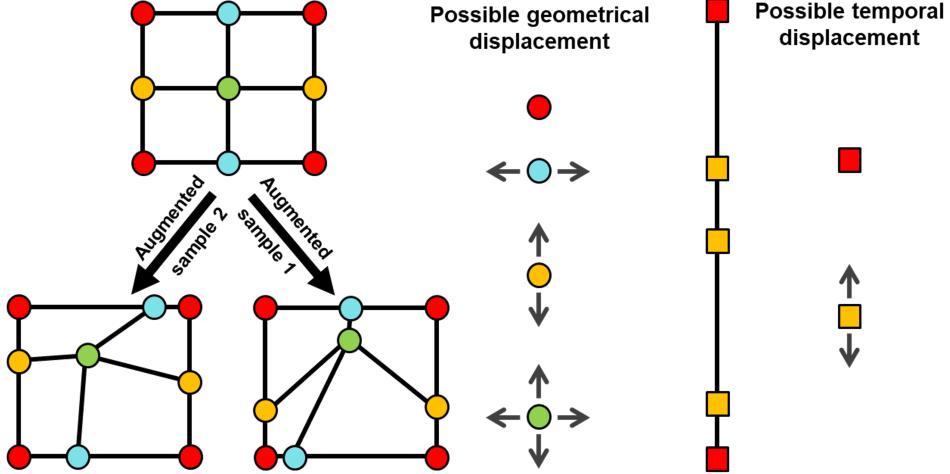


Figure 5.2: Sample of data augmentation in the spatial and temporal domains, with the points constrained to move on possible or allowable directions.

edges is straightforward in a simple, small-scale model used for DA, as exemplified in the subsequent subsection.

After the spatiotemporal perturbation, it is necessary to randomly rotate the coordinate axes using a randomly generated rotation matrix  $\mathbf{R}$ , as they might differ across scales. In this way, any component of data  $\mathbf{T}$  can be rotated to  $\mathbf{T}_{\text{rotated}}$ , utilizing finite strain theory [Holzapfel 2000], as follows:

$$\mathbf{T}_{\text{rotated}} = \begin{cases} \mathbf{R}^T \cdot \mathbf{T} \cdot \mathbf{R} & \text{if } \mathbf{T} \text{ is a symmetric 2nd-order tensor,} \\ \mathbf{T} \cdot \mathbf{R} & \text{if } \mathbf{T} \text{ is a 1st-order tensor or a vector,} \\ \mathbf{T} & \text{if } \mathbf{T} \text{ is a scalar or one-hot encoding.} \end{cases} \quad (5.4)$$

These sequential transformations can create new interpolated nodal data with perturbed element configuration during training, as exemplified in Fig. 5.3. However, the use of such on-the-fly DA and as well as complex subgraphing may bottleneck training. One remedy is our hybrid implementation, as inserting a solver in the surrogate can significantly accelerate training. This is achieved by simplifying the learning problem, as illustrated in the previous chapter, and by facilitating the transfer of boundary information to central areas, potentially reducing the need for high message passing [Um et al. 2020].

We further accelerate training using a custom training loop, depicted in Fig. 5.4. It implements each epoch of training with preprocessing and an early stopping algorithm, followed by another nested loop for training, also employing another early stopping. This process ensures new data is introduced only when necessary, thus speeding up training. Nonetheless, this incremental learning approach risks overshooting. To mitigate this, the algorithm decreases the learning rate,  $\alpha$ , by a factor of 10 at the end of each epoch, and

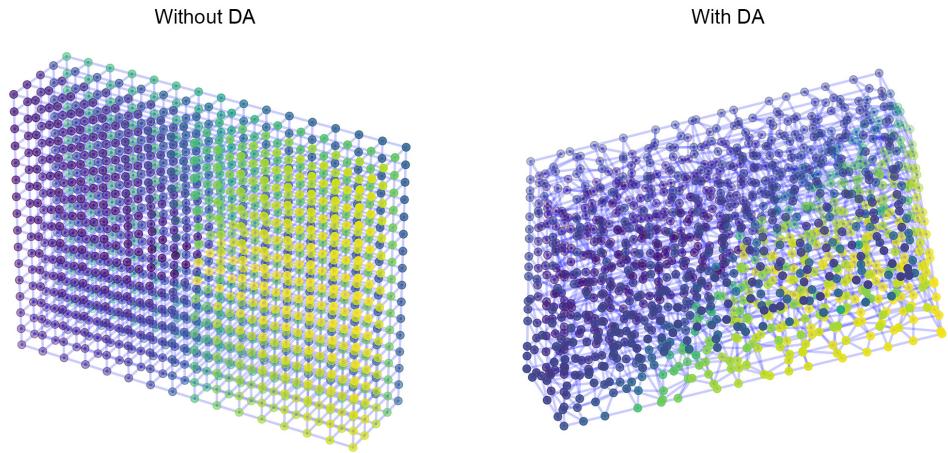


Figure 5.3: Sample of a model with and without data augmentation.

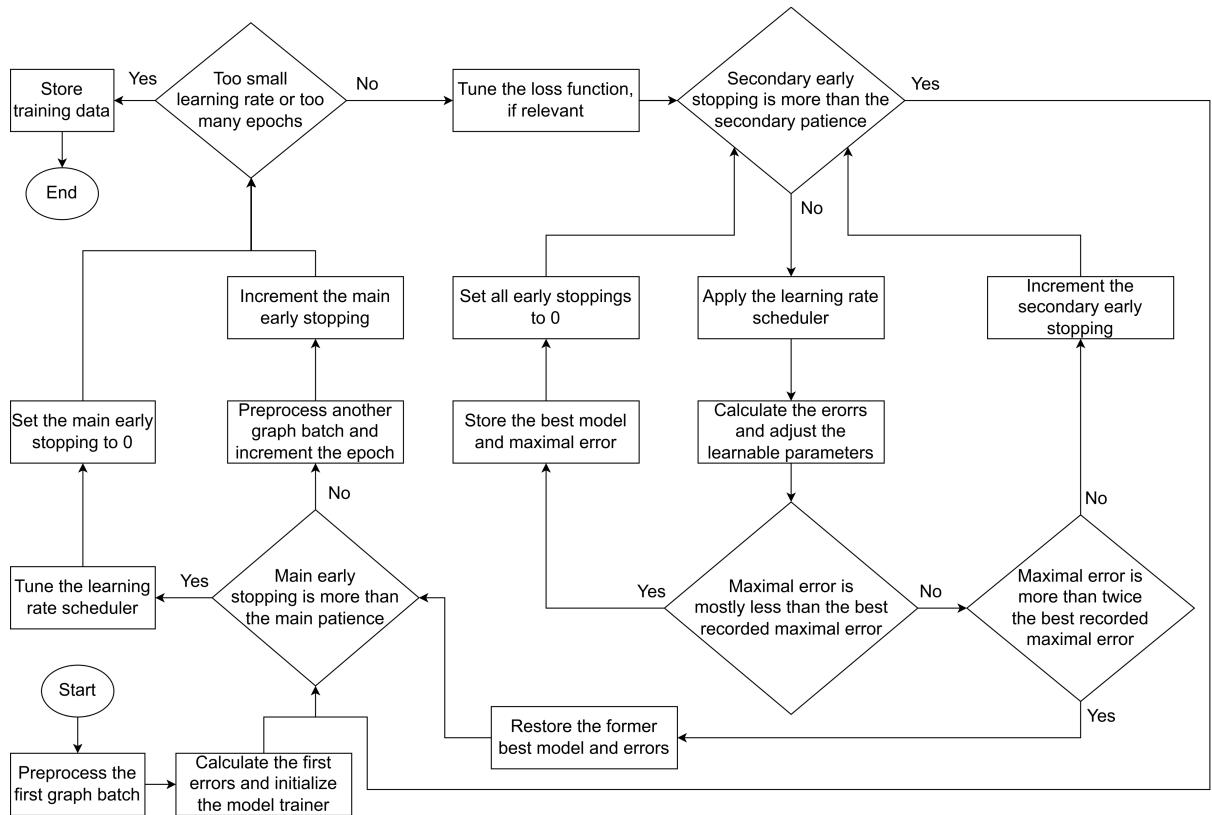


Figure 5.4: Key stages of the custom training loop. The primary distinction from standard training loops is the incremental preprocessing of large batches of graphs at each epoch, coupled with iterative training within a separate inner loop for these large batches; thereby speeding up the training process. Additionally, the early stopping algorithms and learning rate schedulers are employed to prevent overshooting and overfitting, respectively.

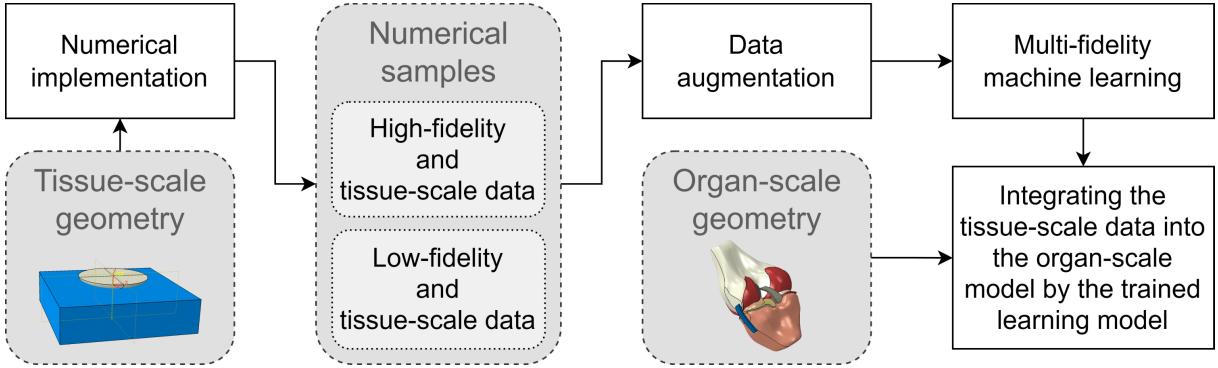


Figure 5.5: Illustration of the role of different models at different scales in surrogate modeling.

then progressively increases it in the nested loop, as follows:

$$\alpha = \frac{1}{10^{i_{\text{outer}}}} \left[ \alpha_{\text{init}} + \min \left( \frac{i_{\text{inner}}}{N_{\text{inner}}}, 1 \right) \cdot (\alpha_{\text{final}} - \alpha_{\text{init}}) \right]. \quad (5.5)$$

Here,  $i_{\text{inner}}$  and  $N_{\text{inner}}$  denote the iteration count and patience number of the secondary (inner) loop, respectively.  $i_{\text{outer}}$  represents the iteration count of the primary (outer) loop, i.e., the epoch number.  $\alpha_{\text{init}}$  and  $\alpha_{\text{final}}$  refer to the initial and final learning rate ranges in the first epoch. The optimal data is stored to be restored if overshooting occurs. Training terminates when the learning rate becomes significantly reduced. This methodology permits efficient implementation of our training experiments, even on a regular central processing unit (CPU).

### 5.2.2 Simulation Tests

We systematically analyze the impact of DA and other proposed methods, focusing on the balance between generalizability and efficiency in complex biomechanical modeling. Initially, datasets are generated through small-scale simulations and subsequently tested in large-scale simulations, as depicted in Fig. 5.5.

**Dataset generation.** We implement the joint-scale model from the previous chapter, now enhanced with a complex multiphasic model for HF modeling of cartilage. This includes all major components, as based on eqs. 2.15 and 2.16. For the small-scale simulation, we employ a rectangular cube with dimensions  $7.5 \times 7.5 \times 1.9 \text{ mm}^3$ . This cube is subjected to compression by an indenter, mimicking the material behavior of soft tissue, here, a meniscus. In modeling cartilage at LF, we assume linear viscoelasticity to capture the dynamic behavior of the tissue through a simpler monophasic model, based on eqs. 2.12 and 2.26. This is implemented using the 8-node linear brick elements, and we set  $\mathcal{G} = 0.744$ ,  $\mathcal{K} = 0.978$ , and  $\mathcal{T} = 13.3 \text{ Sec}$ . Additionally, Young's modulus and Poisson's ratio are set at  $37.8 \text{ MPa}$  and  $0.47$ , respectively. These values, adapted from former

cartilage research [Keenan et al. 2013], are modified to enhance numerical convergence and to increase elasticity, thereby enriching the global data in the nodes, facilitating the subsequent HML analyses.

The indenter used in the model is semi-elliptic with radii of 2 and 0.5 mm. We apply random displacement boundary conditions at a central reference point, kinematically coupled to the flat surface of the indenter. The bottom of the model is fully fixed. By approximating the normalized depth from the top and bottom surfaces, it effectively mimics a simple cut of the tissue. This approach, which employs a few elements of highly structured shapes, accelerates data generation. Ideally, we should have allowed fluid diffusion on the free-contacting surfaces for better fidelity, but this is omitted to simplify the numerical implementation. For increased accuracy, the indenter could have been designed using cartilage material and various shapes to better simulate the effect of complex boundary conditions. While these assumptions may compromise biomechanical fidelity, we intend to design reproducible physics problems by complex multiphasic cartilage models with generalizable surrogate modeling, rather than creating a validated application-oriented biomechanical model of cartilage, which lies beyond the scope of this ML study.

By applying randomly sampling from uniform distribution ( $U$ ), we sample axial displacements of indented from  $U(0, 0.6 \text{ mm})$ , whereas planar displacements are sampled from  $U(0.5, 0.5 \text{ mm})$ , and rotational displacements are sampled from  $U(-0.1, 0.1 \text{ rad})$ . In our large-scale analysis, human weight is sampled by  $U(0, -1000 \text{ N})$ . For all scenarios, the simulation time is sampled between  $U(0.001, 100 \text{ Sec})$  to simulate different strain rates. Additionally, the rate of boundary conditions in small-scale simulations is further randomized with four weighting multipliers that are equally spaced in time, sampled from  $U(0, 1)$ . These weights are interpolated over the entire time range using a cubic spline interpolator [De Boor 1978], to minimize extreme variations. During implementation, we begin with the HF models, recording also the time points for temporal synchronization between models of different fidelities (Fig. 5.6). The HF model outputs include fluid pressure and the Mises stresses of solid components, which serve as nodal target values. The stress-strain results, derived by the LF model, along with the primary fibrillar directions, are used as input nodal features and are stored for use as nodal features. We also store time and one-hot encoded data, distinguishing surface nodes from others, as additional input nodal features.

We also identify the allowable edges for each node for DA in the small-scale model. These could be manually preselected and stored due to the simplicity of its geometry. However, we develop a simple algorithm that aggregates the indices of each connected edge and then ranks them by their quantity. Nodes with fewer edges (e.g., corner nodes) are considered fully fixed during DA (i.e., all their edges are non-allowable). Conversely, nodes with the most edges (e.g., inner nodes) have all their edges set as allowable for DA to enable nodal movements on them. For other nodes, edges connected to higher-ranked

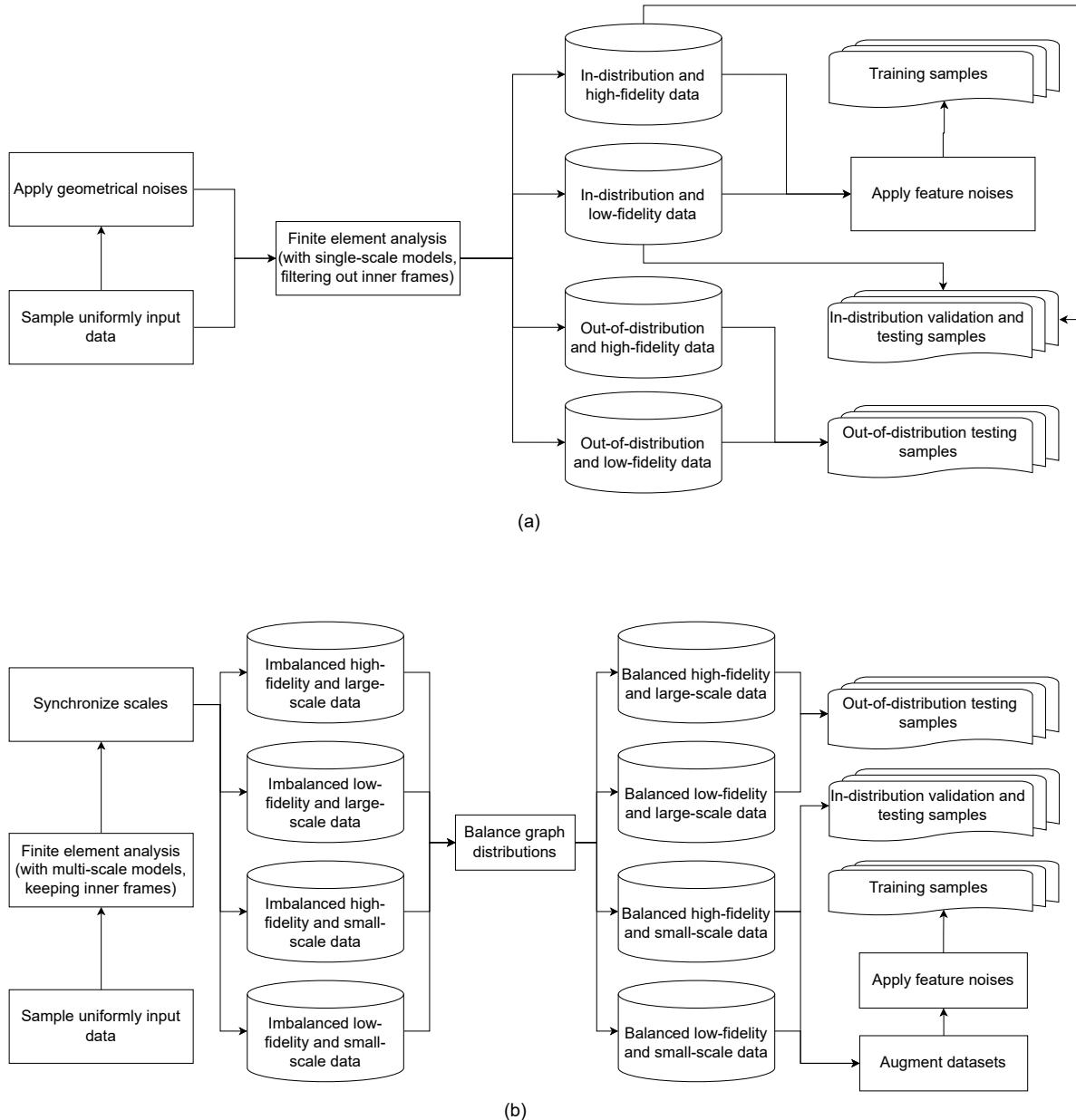


Figure 5.6: Comparison of data generation and preprocessing pipelines for hybrid machine learning (a) and its extended version (b). For both methods, numerical sampling is executed by independent uniform distributions (considering that the input spaces are not extremely large). Furthermore, the extended version does not disregard the inner temporal frames, thus requiring synchronization through temporal interpolation. It also augments the dataset with geometric noises after data generation, allowing for training samples to be gathered from simple and structured meshes.

nodes are assumed non-allowable to maintain boundary shape unchanged. We store this nodal set of allowable edges for random selection during DA preprocessing.

We generate 50 small-scale samples, divided into three roughly equal groups: training, validation, and in-distribution testing samples. Out-of-distribution samples are generated from six far more resource-intensive large-scale simulations. To expedite the simulation process, we applied an arbitrarily large value for the maximum allowable change in fluid pressure, a method occasionally employed in similar cartilage studies (see, e.g., the Mononen et al. 2023 study).

**Training experimentation.** We first extracted the preprocessing hyperparameters used for normalization and Yeo-Johnson transformations, based on samples from the graphs generated for each numerical sample in the training set. The edge features, also normalized, are formulated as described in the previous chapter. In our on-the-fly pre-processor, we create graphs across all temporal frames for subsequent frame sampling. Our experiments include sampling one and two randomly selected closely located or interpolated frames (for 2 and 5 message passings respectively), which are concatenated to augment nodal and edge features. Then, DA is applied to these graph frames, with their relative temporal distances added as additional nodal input feature values. All these graphs are combined using disjoint unions of 100 sampled frames to create a large discrete graph batch for training.

We also experiment with application of different loss functions and subgraph resamplings, as will be reported in the following section. Additionally, a small Gaussian noise with a standard deviation of 0.01 was added to the nodal features. For further exploration, we experimented with models where an upstream AE with a single compressive layer was used to encode input features. A compression ratio (CR) is utilized to report the ratio of the original input nodal features to the output size of the first latent layer. The primary and secondary patience hyperparameters are set to 2 and 100 respectively. Training commences with an initial learning rate of 0.1, and is stopped when it reduces to less than  $10^{-5}$ . Finally, for resampling, we employed 1000 subgraphs summed in each batch. To obtain sample or resampling weights, we used 50 bins for dynamic implementation, while 2 bins were employed for static implementation to avoid the curse of dimensionality. Other hyperparameters remained consistent with those used in the previous chapter.

For the evaluation, we also consider scenarios excluding our proposed methods, all evaluated by MSEs and PMSEs, applied to transformed targets. This approach ensures a fair comparison when averaging all outputs. It should be noted that the targets might be imbalanced and have highly varied ranges, which can significantly impact both learning and evaluation, as demonstrated in previous studies. However, with the preprocessing transformation, the targets' ranges are transformed into similar normalized distributions, and given that we also employ a cubic transformation, the error is highly influenced by

even small potential nodal errors. Consequently, we set the training settings to achieve an error between than 0.1 and 0.001, as greater accuracy might be excessively costly and unnecessary, considering the inherent errors in numerical models which are, by nature, approximate.

Furthermore, while the in-distribution testing batch size remains 100, the out-of-distribution batch size is reduced to 5 since it hardly fits into memory due to the large scale of the knee model and complex preprocessing. This might influence the evaluation results from sampling; however, in most cases when interpreting the results, we focus on the overall pattern and disregard outliers, instead of considering a single sample that can be too simple or complex and not representative of the overall accuracy and generalizability.

Obviously, the hyperparameters, training settings, and even FE modeling might be further modified to enhance performance. In particular, in our experiments, we set them in a way to ensure overall fair comparisons, as in large batch sizes they all trained efficiently. Although we do not disregard the possibility that they might outperform with smaller batches, this makes experimentation infeasible under such different conditions. Yet, as will be shown, we still achieved sufficiently good accuracy with our extension.

While we have explained the implementation steps, interested readers can refer to the open-sourced repository<sup>1</sup> for further details.

### 5.3 Results and Discussion

In order to evaluate the generalizability of the extended HML approach in cartilage biomechanics, a series of simulation tests were performed. These tests utilized varying fidelities and scales and were executed on a regular computing setup (a Core i5 CPU). Detailed runtime metrics are presented in Tables A.3 and A.4 in Appendix A, with a summary provided in Table 5.1.

Table 5.1: Runtimes (Min) with their mean and standard deviation for each model.

	Small-scale simulation	Large-scale simulation
High-fidelity model	$21.21 \pm 2.16$	$118.31 \pm 6.42$
Low-fidelity model	$1.31 \pm 0.69$	$3.07 \pm 0.27$

A notable observation is the significant reduction in runtime compared to the efficiency gains recorded in the previous chapter, as, here, all the main components of cartilage are included in the simulation. This results in relative differences in runtime ranging from 15 to 38 times across different fidelities. Furthermore, the large-scale simulations exhibited slower performance, approximately 2 to 5 times, rendering the small-scale models more

---

<sup>1</sup><https://github.com/shayansss/ehml>.

favorable choices for training. This discrepancy in runtime could potentially be amplified with the implementation of more complex boundary conditions in the knee models. Overall, the LF models are efficient in all scales, justifying the applicability of the generalizable and hybrid machine learning framework of this study.

In this section, we report a series of comparative analyses to evaluate the effectiveness of extended HML with and without the proposed extensions. The evaluations are performed across different simulation scales, while we use small-scale samples for training purposes. The ML models are trained on a GPU, while in parallel, preprocessed on a CPU in a few hours. Nonetheless, they could have been implemented sequentially on a regular CPU.

**Tissue-scale simulation.** Prior to the training phase, a comprehensive preprocessing analysis is conducted on the nodal data of both inputs and outputs. This analysis is executed using various transformations, both individually on the ablated pipeline and in combination. The observed imbalance and skewness support our first hypothesis about the existence of such data, as shown in Fig. 5.7. We addressed this using a series of transformations, as such data might have led to biased training and reduced performance [Rezvani et al. 2023].

As expected, the normalization transformation diminished well data imbalance, yet it fell short in rectifying the skewness inherent in the data. Conversely, the Yeo-Johnson transformation demonstrates a marked improvement in centering the data, especially for the input features. Its combination with the normalization transformation could then potentially normalize the dataset and mitigate skewness. However, a notable imbalance persists, predominantly around the zero value, likely representing unaffected cartilage regions. The introduction of a cubic transformation, in conjunction with the aforementioned methods, shows potential in further refining the dataset. This approach not only addresses the remaining imbalances but also contributes to a more uniform data distribution, thereby preparing it for effective training and fair evaluation.

Figures 5.8 and 5.9 elucidate the training validation MSEs, with and without AE. Likewise, the testing MSEs are averaged and detailed in Tables B.1 and B.2 in Appendix B. It is observed that, in a majority of instances, a single graph batch (implemented repeatedly in one training epoch) is sufficient to attenuate the errors to negligible magnitudes. Exceptions are the models with lower CR and more complexity, where feeding more training batches are still beneficial. Intriguingly, the integration of an upstream AE did not yield an enhancement in training. In contrast, conventional loss functions exhibited superior performance, predominantly in complex models with more message-passing steps. Nevertheless, it is imperative to note that such enhancements do not imply improved generalization capabilities (as will be discussed). On a broader spectrum, the testing (in-distribution) errors predominantly align within comparable ranges, thereby supporting

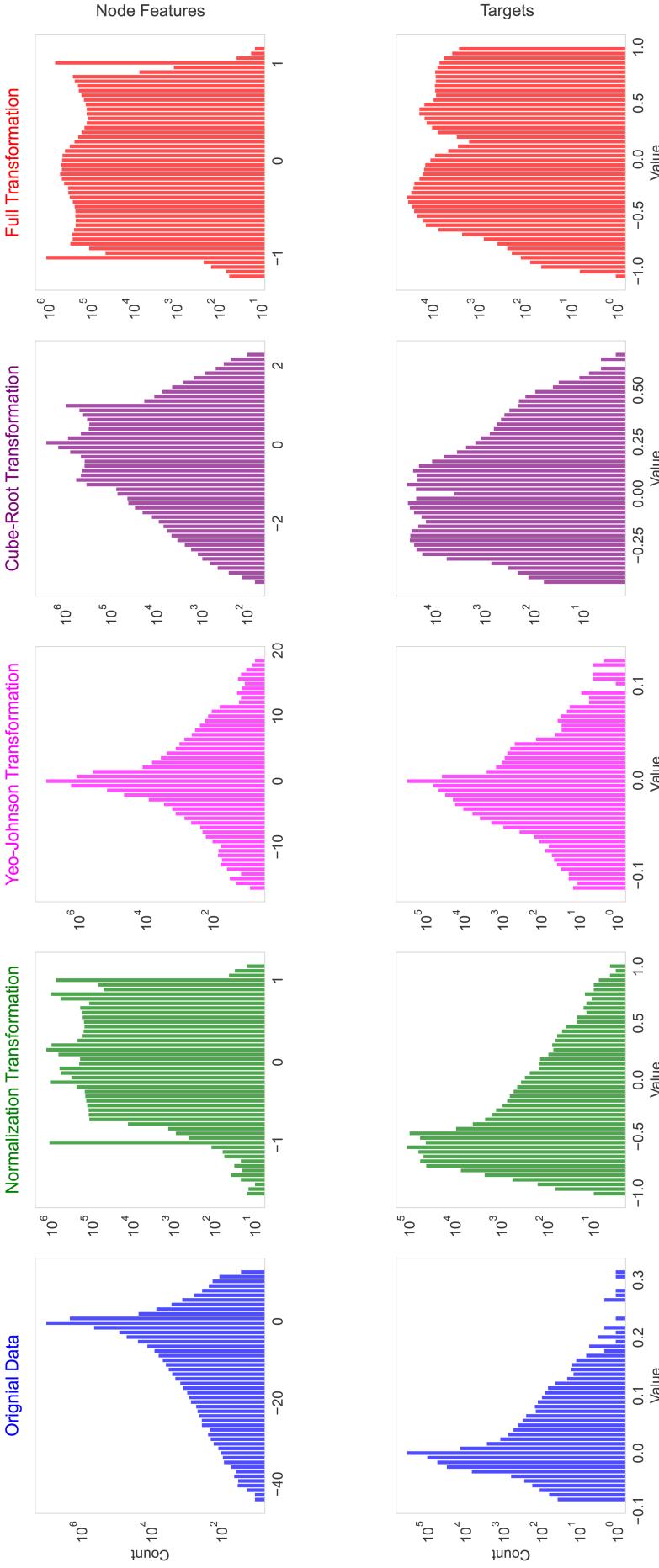


Figure 5.7: Comparison of various preprocessing transformations on nodal input and output features. These transformations were applied to each feature dimension independently but were aggregated here solely for illustration purposes. Overall, the transformations alone could provide a more standardized normalized distribution, while their combination yielded a more uniform distribution. This approach can make the average evaluation errors highly sensitive to a small minority of complex nodes, rather than the majority of simpler nodes, which are typically in areas far from the boundaries (having small values). The existence of many small but few large values is attributable to the nature of soft tissues, where only small regions deform and resist pressure significantly more than other areas, particularly when the deformation is moderate but applied rapidly, as implemented in this study.

our hypothesis that the application of HML allows opting for simpler models, allowing faster training [Tan et al. 2021].

**Knee-scale simulation.** In Fig. 5.10, we present a comprehensive analysis of the out-of-distribution testing errors, as detailed in Tables B.4 to B.10 in Appendix B. We acknowledge the limited sample size employed in this knee-scale simulation, a constraint imposed by the infeasibility of storing an extensive array of snapshots in memory. This limitation might introduce some bias in the error metrics due to sampling variability. Despite this, a discernible pattern emerges from the aggregated data, disregarding the outliers. Remarkably, the application of more complex models, particularly with AE, does not again manifest in substantial performance enhancements. Conversely, DA exhibits a pronounced positive impact, primarily because models devoid of augmentation lack exposure to diversified element configurations. Notably, the models implementing a weighted loss approach consistently demonstrate the most minimal error margins, coupled with a markedly lower variance.

To further assess the efficacy of our novel graph-based DA technique, we examined the averaged pointwise errors, as depicted in Fig. 5.11. Notably, the regular loss function yielded the highest errors, while the implementation of weighted loss functions resulted in significantly reduced errors, almost negligible in comparison. Although maximal loss may still hold value in simpler models, it demands very large training batch sizes. Interestingly, subgraph-based methods did not surpass the performance of weighting strategies, likely due to their specific tuning requirements. Dynamic versions of these methods, requiring minimal tuning, showed superior performance over their static counterparts with DA. Regardless of the approach used for addressing data imbalance, the application of our DA algorithm consistently achieved a substantial reduction in errors, further validating its effectiveness.

**Other discussion.** We acknowledge two key limitations. First, the subgraph implementation might be too specific. Indeed, we updated the dynamic version at each training epoch instead of each step, a decision influenced by potential performance bottlenecks. This approach, while effective for our objectives, leaves room for future optimization in different training scenarios [Xue et al. 2023]. Second, although we used a complex multi-phasic model (with pre-stressing) for the large-scale simulation, the numerical implementation and imposed boundary conditions are again very specific to avoid very demanding HF data generation. However, this constraint aligns with the methodological nature of this work, as experiments are designed to evaluate the effectiveness of our extensions, particularly the DA approach, rather than executing diverse HF simulations and biomechanical analyses.

To summarize, we proposed and empirically tested an extended HML framework tai-

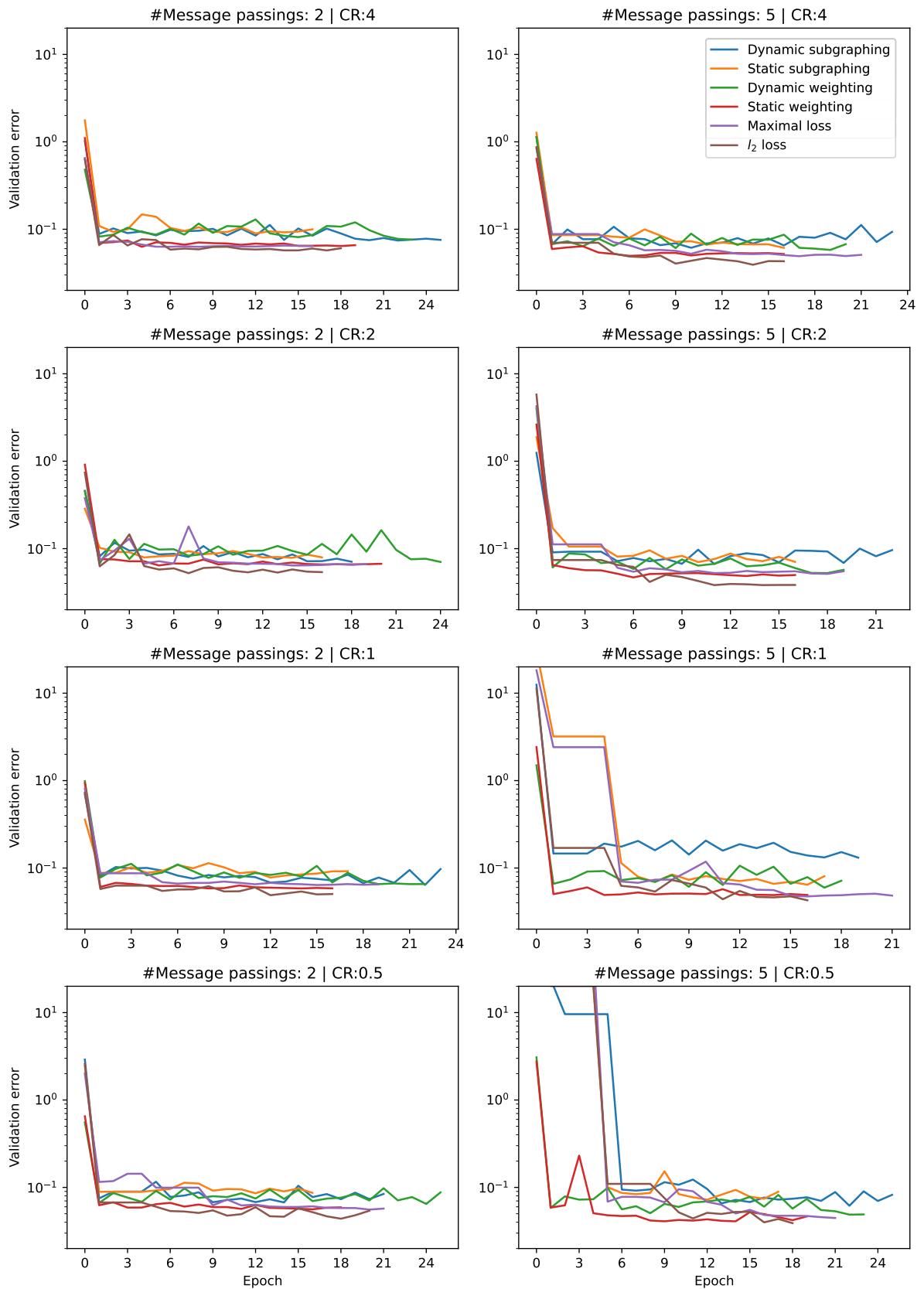


Figure 5.8: Comparative analysis of validation errors across training epochs for the models without the autoencoder. Abbreviation: CR = compression ratio.

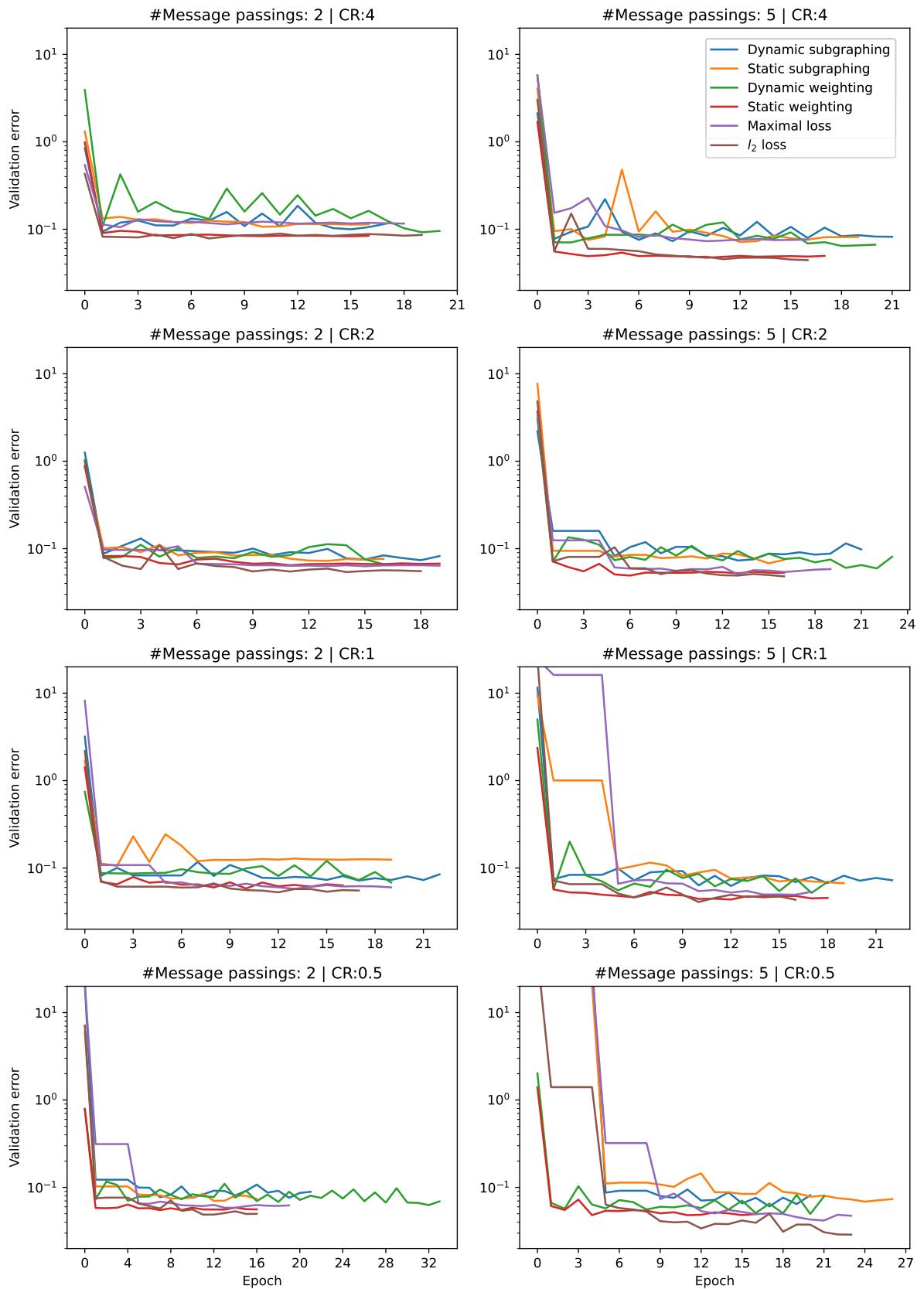


Figure 5.9: Comparative analysis of validation errors across training epochs for the models with the autoencoder. Abbreviation: CR = compression ratio.

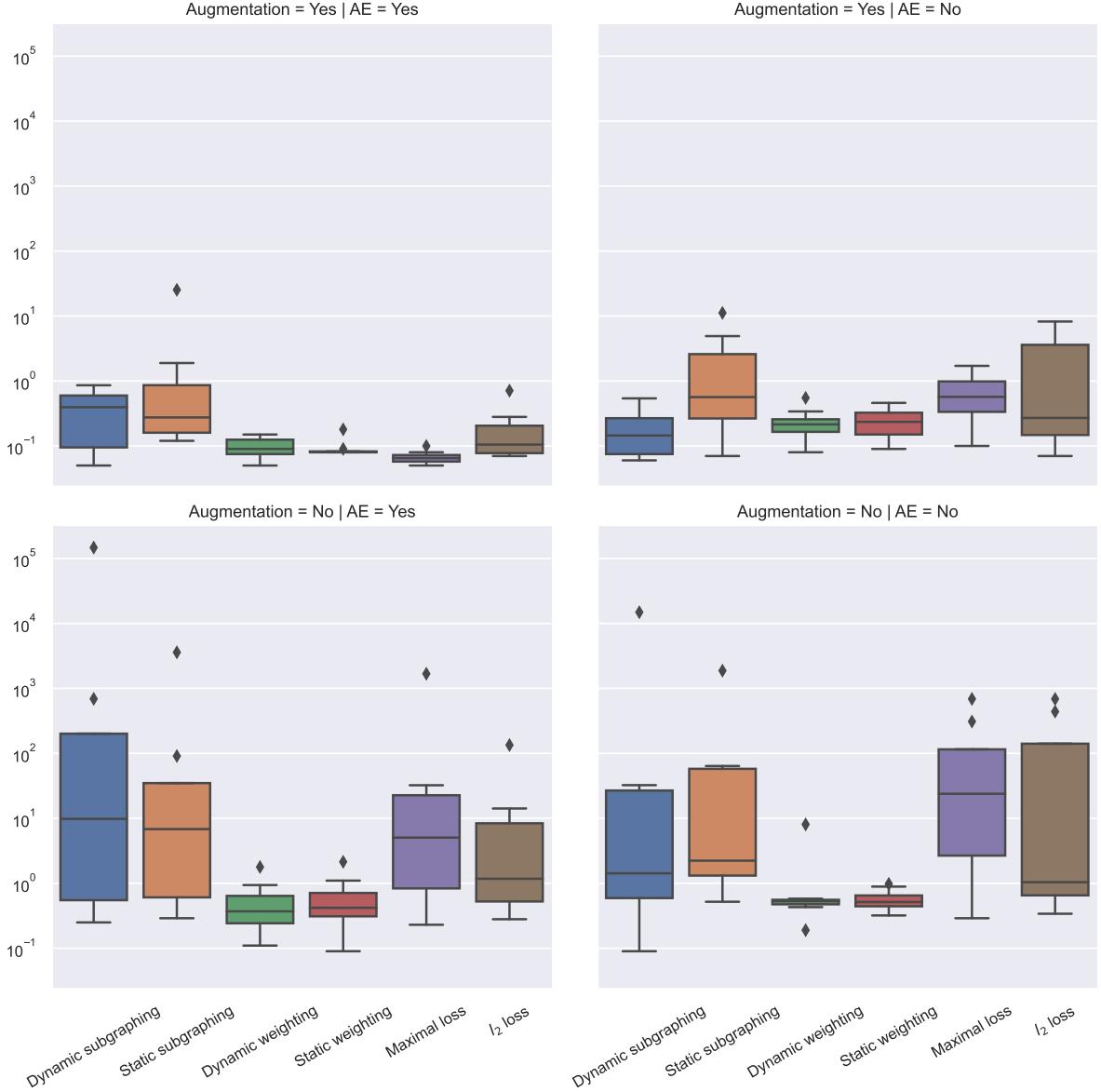


Figure 5.10: Averaged out-of-distribution testing errors for various models trained using different loss functions. Abbreviation: AE = autoencoder.

lored for multiphasic surrogate modeling of knee cartilage, showcasing potentials in both few-shot learning and zero-shot generalization. This approach integrates a customized training loop and a hybrid (multi-fidelity) scheme, permitting efficient training. These models exhibited sufficient accuracy even with a minimal number of message-passing blocks and without AE-based MOR. Additionally, we conducted a comparative analysis of various preprocessing, cost-sensitive, and resampling strategies, underscoring their critical role in addressing data imbalance and skewness that impacted training. Most importantly, the chapter also highlights how our proposed physics-based DA technique significantly improves the surrogate model's generalizability. Such advancements might also mitigate the prevalent issues of limited generalizability and efficiency in existing ML-

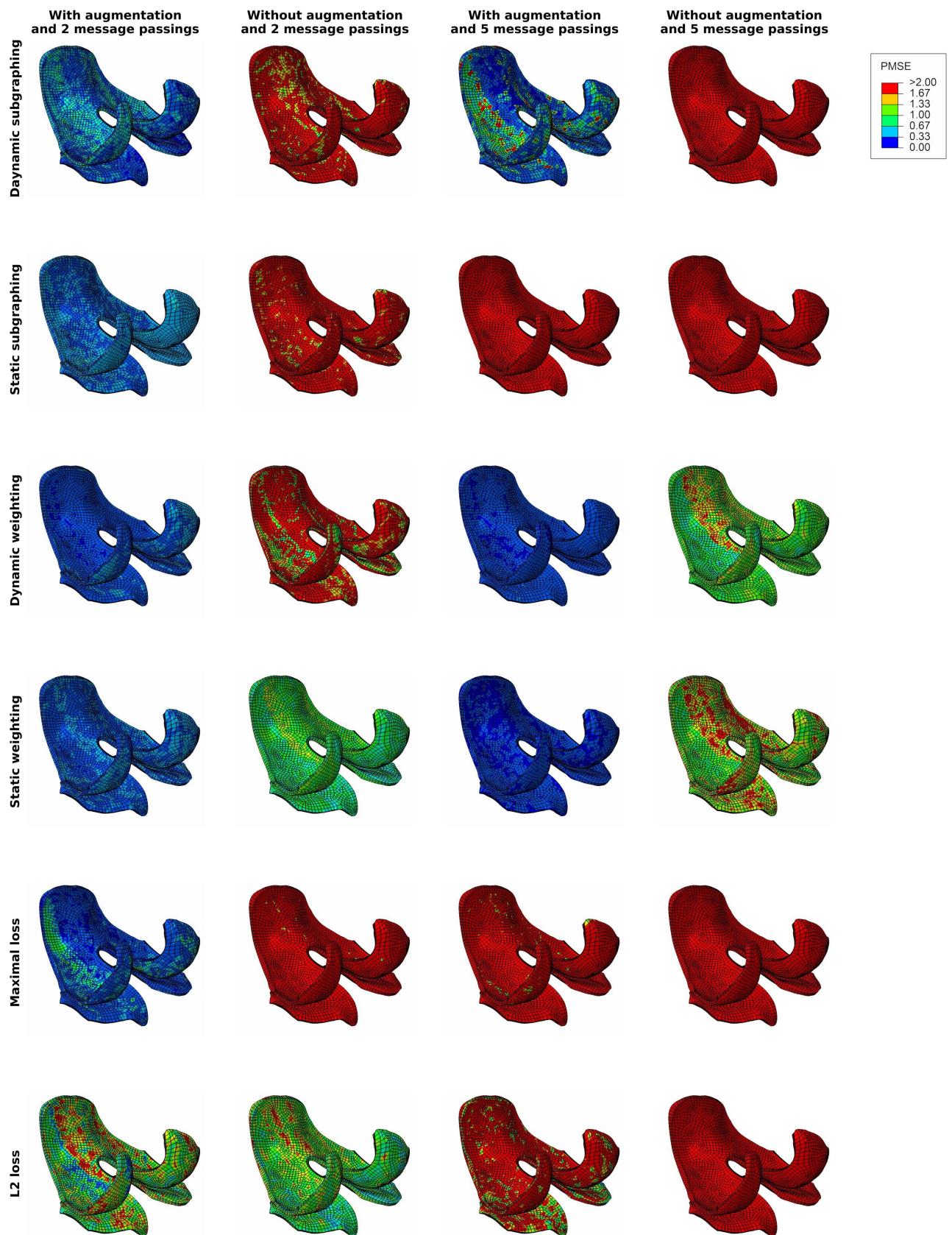


Figure 5.11: Averaged (out-of-distribution testing) pointwise mean-squared errors (PMSEs) for various models trained using different loss functions.

based surrogate models, potentially extending their applicability across broader multiscale domains.

# Chapter 6

## Conclusion

This work is one of the first attempts to provide a multi-physics numerical and data-driven model of knee cartilage with implementation efficiency. At the heart of this study was the development of an efficient, generalizable, and scalable surrogate modeling framework, trained on a multiphasic model of cartilage across various scales and fidelities. Then, surrogate models were developed with the assumption of limited training data, as HF biomechanical models might be too expensive to generate the data. By integrating state-of-the-art ML and FE analysis, we sought to overcome these challenges and developed four research questions in this regard.

Addressing the first research question, “*How can a complex multi-physics model of cartilage biomechanics be implemented efficiently enough to generate the required training datasets?*”, we successfully implemented a complex multi-physics model of cartilage biomechanics with a fully automated PSA, optimized for efficient subsequent HF dataset generation. This model was designed with a focus on balancing computational efficiency with the numerical fidelity necessary to capture the heterogeneous and multiphasic nature of cartilage tissue. Besides, the key to efficiently generating the training datasets was found in the strategic simplification of our multi-physics cartilage model, used subsequently in a HML method.

The second research question, “*How can machine learning techniques be integrated into the numerical model for efficient surrogate modeling?*”, was addressed through the development of a novel hybrid multi-fidelity learning framework. It combines the LF (reduced-order) numerical analysis with model-agnostic supervised ML techniques. The cornerstone of this integration was the creation of surrogate models that are trained on a limited number of HF numerical samples. The surrogate models developed showcased a significant reduction in computational time compared to traditional numerical methods, achieving the goal of efficient surrogate modeling.

In light of the third research question, “*Is the new machine learning system scalable and generalizable across different scales?*”, we expanded the scope of our study to encompass multiscale simulations. The scalability and generalizability of our HML framework were

rigorously tested by employing it in both tissue-scale and knee-scale simulations. We tested the integration of an upstream AE model, along with the application of various preprocessing and customization algorithms, to evaluate their effects on performance. Our findings suggest that the augmented HML framework, combined with GNNs, can effectively learn from small-scale data and generalize its knowledge to larger, more complex simulations, thereby answering positively this question.

Addressing the final research question, “*What are the potential implications of the proposed methods for future research?*”, we believe that our proposed methods can significantly contribute to future development due to their non-intrusive implementation of advanced cartilage biomechanics. Our HF pre-stressed and multi-physics cartilage model can be efficiently implemented since it is based on a combination of regular nonlinear FE and a FPBBS. The data-driven extensions are also based on widely adopted deep learning (DL) algorithms, again with efficient execution and fast training.

Nonetheless, during our evaluation, we designed the simulations specifically to efficiently generate datasets. We recognize this as a limitation but have strived to enhance the generalizability of our results by implementing complex biomechanical models in various scenarios. However, for application-oriented studies, further research may be needed to devise the best strategies for generating the most informative training samples corresponding to each biomechanical simulation. Since sample-wise validation of the model may not be generalizable or even feasible, especially in large-scale testing where we may have zero testing samples in practice, we may resort to further available evaluation errors. However, these metrics also have their limits [Barocas et al. 2023] and therefore cannot replace rigorous *in vivo* or *in vitro* validation. Such tests, however, are highly time-consuming and problem-specific and thus remain for future studies.

Among possible future directions to advance our method is the integration of generative models [Rezende et al. 2014; Kingma et al. 2014], to replace the preprocessing steps. This integration could accelerate preprocessing tasks and potentially eliminate the need for our custom training loops. Furthermore, applying our method to more advanced, simulation-specific analyses, especially with more complex boundary conditions [Lenhart et al. 2015], is another interesting research direction. In particular, our hybrid method has significant potential to expedite the repetitive process of cartilage analysis, e.g., [Elahi et al. 2021], particularly when incorporating osmotic pressure, as in [Elahi et al. 2023]. Also, since it has been successfully tested for zero-shot generalization, it might be applied for very interesting multiscale modeling with even non-iterative tasks at large-scale, while trained on small-scale data.

More specifically, we anticipate the use of our methods in the context of material calibration, as conventional learning approaches may require thousands of training samples [Arbabi et al. 2016b]. This method could substantially reduce the number of samples needed to just a few or could be used to efficiently augment the dataset. Another exam-

ple is the application of advanced constitutive modeling in the personalized simulation of knee osteoarthritis. Regular FE methods, in contrast to our multi-scale method, might have to sacrifice constitutive complexity in large-scale analyses, limiting their capabilities [Mononen et al. 2023]. Moreover, given the similarities in constitutive modeling of soft tissues [Khaniki et al. 2023], the implications of this research can extend far beyond cartilage biomechanics.

On the other hand, while ML studies have embraced the open-source culture, e.g., [Panfilov et al. 2019; Desai et al. 2019; Desai et al. 2021; Thomas et al. 2021; Panfilov et al. 2022], numerical studies in biomechanics are rarely open-sourced, likely due to legal complexities, which complicates the future testing and advancement of our methods. Recently, a few biomechanical or numerical studies have started to democratize the field by sharing their code, e.g., [Fehervary et al. 2020; Maes et al. 2023; Chokhandre et al. 2021], but the scarcity of such models remains a major challenge for future progress. As a remedy, we have open-sourced most of our data, accompanied by detailed documentation.

To conclude, we summarize our main original contributions, which were empirically tested throughout this research, as follows:

1. Development of a cartilage model, optimized for automatic generation of HF training data, with its merits demonstrated by pre-stressing simulation to assess its impact on constitutive behavior and tissue biomechanics.
2. Development of straightforward HML in the context of soft tissue multi-physics, enabling multi-fidelity surrogate modeling of cartilage regardless of the ML model employed.
3. Extension of the hybrid model with a set of relatively efficient algorithms, particularly physics-based DA, permitting multiscale surrogate modeling, implemented by using GNNs.

Furthermore, our research may transcend its initial biomechanical application, showcasing a harmonious blend of learnable AI with traditional numerical analysis, while emphasizing the balance between efficiency and complexity under limited resources.

# References

- Abadi, M., A. Agarwal, P. Barham, et al. (2015). “TensorFlow: Large-scale machine learning on heterogeneous systems”. In: Software available from tensorflow.org. URL: <https://doi.org/10.48550/arXiv.1603.04467>.
- Abaqus (2021). *Dassault Systèmes Simulia Corp., Providence, RI, USA*. URL: <http://www.simulia.com>.
- Abiodun, O.I., A. Jantan, A.E. Omolara, et al. (2019). “Comprehensive review of artificial neural network applications to pattern recognition”. In: *IEEE Access* 7, pp. 158820–158846. URL: <https://doi.org/10.1109/ACCESS.2019.2945545>.
- Ahn, J.G., H.I. Yang, and J.G. Kim (2022). “Multi-fidelity meta modeling using composite neural network with online adaptive basis technique”. In: *Computer Methods in Applied Mechanics and Engineering* 388, p. 114258. ISSN: 0045-7825. URL: <https://doi.org/10.1016/j.cma.2021.114258>.
- Alastrué, V., E. Peña, et al. (2007). “Assessing the use of the “Opening Angle Method” to enforce residual stresses in patient-specific arteries”. In: *Annals of Biomedical Engineering* 35, pp. 1821–1837. URL: <https://doi.org/10.1007/s10439-007-9352-4>.
- Almeida, E.S. and R.L. Spilker (1997). “Mixed and penalty finite element models for the nonlinear behavior of biphasic soft tissues in finite deformation: Part I—alternate formulations”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 1.1, pp. 25–46. URL: <https://doi.org/10.1080/01495739708936693>.
- Alunni Cardinali, M., M. Govoni, M. Tschon, et al. (2023). “Brillouin–Raman micro-spectroscopy and machine learning techniques to classify osteoarthritic lesions in the human articular cartilage”. In: *Scientific Reports* 13.1, p. 1690. ISSN: 2045-2322. URL: <https://doi.org/10.1038/s41598-023-28735-5>.
- Antman, S.S. and J.E. Osborn (1979). “The principle of virtual work and integral laws of motion”. In: *Archive for Rational Mechanics and Analysis* 69.3, pp. 231–262. ISSN: 1432-0673. URL: <https://doi.org/10.1007/BF00248135>.
- Arbabi, V., B. Pouran, G. Campoli, et al. (2016a). “Determination of the mechanical and physical properties of cartilage by coupling poroelastic-based finite element models of indentation with artificial neural networks”. In: *Journal of Biomechanics* 49.5, pp. 631–637. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2015.12.014>.

- Arbabi, V., B. Pouran, H. Weinans, et al. (2016b). “Combined inverse-forward artificial neural networks for fast and accurate estimation of the diffusion coefficients of cartilage based on multi-physics models”. In: *Journal of Biomechanics* 49.13, pp. 2799–2805. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2016.06.019>.
- Ardatov, O., V. Aleksiuk, A. Maknickas, et al. (2023). “Modeling the impact of meniscal tears on von Mises stress of knee cartilage tissue”. In: *Bioengineering* 10.3. ISSN: 2306-5354. URL: <https://doi.org/10.3390/bioengineering10030314>.
- Astephen Wilson, J.L. and D. Kobsar (2021). “Osteoarthritis year in review 2020: Mechanics”. In: *Osteoarthritis and Cartilage* 29.2, pp. 161–169. ISSN: 1063-4584. URL: <https://doi.org/10.1016/j.joca.2020.12.009>.
- Ateshian, G.A., N.O. Chahine, I.M. Basalo, et al. (2004). “The correspondence between equilibrium biphasic and triphasic material properties in mixture models of articular cartilage”. In: *Journal of Biomechanics* 37.3, pp. 391–400. ISSN: 0021-9290. URL: [https://doi.org/10.1016/S0021-9290\(03\)00252-5](https://doi.org/10.1016/S0021-9290(03)00252-5).
- Backus, G. (1970). “A geometrical picture of anisotropic elastic tensors”. In: *Reviews of Geophysics* 8.3, pp. 633–671. URL: <https://doi.org/10.1029/RG008i003p00633>.
- Barocas, S., M. Hardt, and A. Narayanan (2023). *Fairness and machine learning: Limitations and opportunities*. MIT Press.
- Battaglia, P.W., J.B. Hamrick, V. Bapst, et al. (2018). *Relational inductive biases, deep learning, and graph networks*. URL: <https://doi.org/10.48550/arxiv.1806.01261>.
- Baydin, A.G., B.A. Pearlmutter, et al. (2018). “Automatic differentiation in machine learning: a survey”. In: *Journal of Machine Learning Research* 18.153, pp. 1–43. URL: <https://doi.org/10.48550/arXiv.1502.05767>.
- Belytschko, T., W.K. Liu, B. Moran, et al. (2014). *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons.
- Benninghoff, A (1925). “Form und bau der Gelenkknorpel in ihren Beziehungen zur Funktion: Zweiter teil: Der Aufbau des Gelenkknorpels in seinen Beziehungen zur Funktion”. In: *Zeitschrift für Zellforschung und mikroskopische Anatomie* 2.5, pp. 783–862.
- Berner, J., P. Grohs, G. Kutyniok, et al. (2022). “The modern mathematics of deep learning”. In: *Mathematical Aspects of Deep Learning*. Ed. by P. Grohs and G. Kutyniok. Cambridge University Press, pp. 1–111. URL: <https://doi.org/10.1017/9781009025096.002>.
- Bhattacharya, S., V. Rajan, and H. Shrivastava (2017). “ICU mortality prediction: A classification algorithm for imbalanced datasets”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1. URL: <https://doi.org/10.1609/aaai.v31i1.10721>.
- Bielajew, B.J., J.C. Hu, and K.A. Athanasiou (2020). “Collagen: Quantification, biomechanics and role of minor subtypes in cartilage”. In: *Nature Reviews Materials* 5.10, pp. 730–747. ISSN: 2058-8437. URL: <https://doi.org/10.1038/s41578-020-0213-1>.

- Birge, J.R. and F. Louveaux (2011). *Introduction to stochastic programming*. 2nd. Springer Publishing Company, Incorporated. ISBN: 1461402360. URL: <https://doi.org/10.1007/978-1-4614-0237-4>.
- Bjorck, N., C.P. Gomes, B. Selman, et al. (2018). “Understanding batch normalization”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, et al. Vol. 31. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2018/file/36072923bfc3cf47745d704feb489480-Paper.pdf>.
- Bols, J., J. Degroote, B. Trachet, et al. (2013). “A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels”. In: *Journal of Computational and Applied Mathematics* 246. Fifth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2011), pp. 10–17. ISSN: 0377-0427. URL: <https://doi.org/10.1016/j.cam.2012.10.034>.
- Bozec, L., G. van der Heijden, and M. Horton (2007). “Collagen fibrils: Nanoscale ropes”. In: *Biophysical Journal* 92.1, pp. 70–75. ISSN: 0006-3495. URL: <https://doi.org/10.1529/biophysj.106.085704>.
- Brody, L.T. (2015). “Knee osteoarthritis: Clinical connections to articular cartilage structure and function”. In: *Physical Therapy in Sport* 16.4, pp. 301–316. ISSN: 1466-853X. URL: <https://doi.org/10.1016/j.ptsp.2014.12.001>.
- Buschmann, M.D. and A.J. Grodzinsky (1995). “A molecular model of proteoglycan-associated electrostatic forces in cartilage mechanics”. In: *Journal of Biomechanical Engineering* 117.2, pp. 179–192. ISSN: 0148-0731. DOI: [10.1115/1.2796000](https://doi.org/10.1115/1.2796000). URL: <https://doi.org/10.1115/1.2796000>.
- Cai, H., V.W. Zheng, and K.C. Chang (2018). “A comprehensive survey of graph embedding: Problems, techniques, and applications”. In: *IEEE Transactions on Knowledge and Data Engineering* 30.9, pp. 1616–1637. URL: <https://doi.org/10.1109/TKDE.2018.2807452>.
- Cai, L., L. Ren, Y. Wang, et al. (2021). “Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium”. In: *Royal Society Open Science* 8.1, p. 201121. URL: <https://doi.org/10.1098/rsos.201121>.
- Cai, S., Z. Mao, Z. Wang, et al. (2021). “Physics-informed neural networks (PINNs) for fluid mechanics: a review”. In: *Acta Mechanica Sinica* 37.12, pp. 1727–1738. ISSN: 1614-3116. URL: <https://doi.org/10.1007/s10409-021-01148-1>.
- Caimi, A., A. Pasquali, F. Sturla, et al. (2020). “Prediction of post-stenting biomechanics in coarcted aortas: A pilot finite element study”. In: *Journal of Biomechanics* 105, p. 109796. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2020.109796>.
- Cao, Y., M. Chai, et al. (2023). “Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, et al.

- Vol. 202. Proceedings of Machine Learning Research. PMLR, pp. 3541–3558. URL: <https://doi.org/10.48550/arXiv.2210.02573>.
- Cederlund, A.A. and R.M. Aspden (2022). “Walking on water: Revisiting the role of water in articular cartilage biomechanics in relation to tissue engineering and regenerative medicine”. In: *Journal of The Royal Society Interface* 19.193, p. 20220364. URL: <https://doi.org/10.1098/rsif.2022.0364>.
- Chan, D.D., C.P. Neu, and M.L. Hull (2009). “Articular cartilage deformation determined in an intact tibiofemoral joint by displacement-encoded imaging”. In: *Magnetic Resonance in Medicine* 61.4, pp. 989–993. URL: <https://doi.org/10.1002/mrm.21927>.
- Charlebois, M., M.D. McKee, and M.D. Buschmann (2004). “Nonlinear tensile properties of bovine articular cartilage and their variation with age and depth”. In: *Journal of Biomechanical Engineering* 126.2, pp. 129–137. ISSN: 0148-0731. URL: <https://doi.org/10.1115/1.1688771>.
- Chen, A., C. Gupte, and K. Akhtar (2012). “The global economic cost of osteoarthritis: How the UK compares”. In: *Arthritis* 2012, p. 698709. ISSN: 2090-1984. URL: <https://doi.org/10.1155/2012/698709>.
- Cheng, M., P. Jiang, J. Hu, et al. (2021). “A multi-fidelity surrogate modeling method based on variance-weighted sum for the fusion of multiple non-hierarchical low-fidelity data”. In: *Structural and Multidisciplinary Optimization* 64.6, pp. 3797–3818. ISSN: 1615-1488. URL: <https://doi.org/10.1007/s00158-021-03055-2>.
- Cheung, H.L., P. Uvdal, and M. Mirkhalaf (2023). *Augmentation of scarce data – a new approach for deep-learning modeling of composites*. eprint: [2311.14557](https://doi.org/10.48550/arXiv.2311.14557). URL: <https://doi.org/10.48550/arXiv.2311.14557>.
- Chiang, W.L., X. Liu, S. Si, et al. (2019). “Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. Anchorage, AK, USA: Association for Computing Machinery, pp. 257–266. ISBN: 9781450362016. URL: <https://doi.org/10.1145/3292500.3330925>.
- Chokhandre, S., E.E. Neumann, T.F. Nagle, et al. (2021). “Specimen specific imaging and joint mechanical testing data for next generation virtual knees”. In: *Data in Brief* 35, p. 106824. ISSN: 2352-3409. URL: <https://doi.org/10.1016/j.dib.2021.106824>.
- Chollet, F. et al. (2015). *Keras*. URL: <https://github.com/fchollet/keras>.
- Clark, J.M. (1985). “The organization of collagen in cryofractured rabbit articular cartilage: A scanning electron microscopic study”. In: *Journal of Orthopaedic Research* 3.1, pp. 17–29. URL: <https://doi.org/10.1002/jor.1100030102>.
- Coutinho, E.J.R. et al. (2023). “Physics-informed neural networks with adaptive localized artificial viscosity”. In: *Journal of Computational Physics* 489, p. 112265. ISSN: 0021-9991. URL: <https://doi.org/10.1016/j.jcp.2023.112265>.

- Cueto, E., F. Chinesta, and A. Huerta (2014). “Model order reduction based on proper orthogonal decomposition”. In: *Separated Representations and PGD-Based Model Reduction: Fundamentals and Applications*. Ed. by F. Chinesta and P. Ladevèze. Vienna: Springer Vienna, pp. 1–26. ISBN: 978-3-7091-1794-1. URL: [https://doi.org/10.1007/978-3-7091-1794-1\\_1](https://doi.org/10.1007/978-3-7091-1794-1_1).
- Culav, E.M., C.H. Clark, and M.J. Merrilees (1999). “Connective tissues: Matrix composition and its relevance to physical therapy”. In: *Physical Therapy* 79.3, pp. 308–319. ISSN: 0031-9023. URL: <https://doi.org/10.1093/ptj/79.3.308>.
- Dai, W. et al. (2007). “Co-clustering based classification for out-of-domain documents”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’07. San Jose, California, USA: Association for Computing Machinery, pp. 210–219. ISBN: 9781595936097. URL: <https://doi.org/10.1145/1281192.1281218>.
- Daumé III, H. (2007). “Frustratingly easy domain adaptation”. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 256–263. URL: <https://doi.org/10.48550/arXiv.0907.1815>.
- De Boor, C. (1978). “A practical guide to splines”. In: *Applied Mathematical Sciences*, p. 392. ISSN: 0066-5452.
- Desai, A.D., M. Barbieri, V. Mazzoli, et al. (2019). “DOSMA: A deep-learning, open-source framework for musculoskeletal MRI analysis”. In: *Proc 27th Annual Meeting ISMRM, Montreal*, p. 1135.
- Desai, A.D., F. Caliva, C. Iriondo, et al. (2021). “The international workshop on osteoarthritis imaging knee MRI segmentation challenge: A multi-institute evaluation and analysis framework on a standardized dataset”. In: *Radiology: Artificial Intelligence* 3.3, e200078. URL: <https://doi.org/10.1148/ryai.2021200078>.
- Deserno, M. and C. Holm (2001). “Cell model and Poisson-Boltzmann theory: A brief introduction”. In: *Electrostatic Effects in Soft Matter and Biophysics*. Ed. by C. Holm, P. Kékicheff, and R. Podgornik. Dordrecht: Springer Netherlands, pp. 27–52. ISBN: 978-94-010-0577-7. URL: [https://doi.org/10.1007/978-94-010-0577-7\\_2](https://doi.org/10.1007/978-94-010-0577-7_2).
- Dong, G. et al. (2022). “A part-scale, feature-based surrogate model for residual stresses in the laser powder bed fusion process”. In: *Journal of Materials Processing Technology* 304, p. 117541. ISSN: 0924-0136. URL: <https://doi.org/10.1016/j.jmatprotec.2022.117541>.
- Duchi, J., E. Hazan, and Y. Singer (2011). “Adaptive subgradient methods for online learning and stochastic optimization”. In: *Journal of Machine Learning Research* 12.61, pp. 2121–2159. URL: <http://jmlr.org/papers/v12/duchi11a.html>.

- Dudhia, J. (2005). “Aggrecan, aging and assembly in articular cartilage”. In: *Cellular and Molecular Life Sciences CMLS* 62.19, pp. 2241–2256. ISSN: 1420-9071. URL: <https://doi.org/10.1007/s00018-005-5217-x>.
- Dullien, F.A.L. (1979). *Porous Media: Fluid Transport and Pore Structure*. Elsevier Science.
- Dye, S.F. (1987). “An evolutionary perspective of the knee.” In: *The Journal of Bone and Joint Surgery. American volume* 69.7, pp. 976–983. ISSN: 0021-9355. URL: <http://europemc.org/abstract/MED/3654710>.
- Ebrahimi, M., S. Ojanen, A. Mohammadi, et al. (2019). “Elastic, viscoelastic and fibril-reinforced poroelastic material properties of healthy and osteoarthritic human tibial cartilage”. In: *Annals of Biomedical Engineering* 47.4, pp. 953–966. ISSN: 1573-9686. URL: <https://doi.org/10.1007/s10439-019-02213-4>.
- Egli, F.S., R.C. Straube, A. Mielke, et al. (2021). “Surrogate modeling of a nonlinear, biphasic model of articular cartilage with artificial neural networks”. In: *PAMM* 21.1, e202100188. URL: <https://doi.org/10.1002/pamm.202100188>.
- Elahi, S.A., R. Castro-Viñuelas, P. Tanska, et al. (2023). “Contribution of collagen degradation and proteoglycan depletion to cartilage degeneration in primary and secondary osteoarthritis: an in silico study”. In: *Osteoarthritis and Cartilage* 31.6, pp. 741–752. ISSN: 1063-4584. URL: <https://doi.org/10.1016/j.joca.2023.01.004>.
- Elahi, S.A., P. Tanska, R.K. Korhonen, et al. (2021). “An in silico framework of cartilage degeneration that integrates fibril reorientation and degradation along with altered hydration and fixed charge density loss”. In: *Frontiers in Bioengineering and Biotechnology* 9. ISSN: 2296-4185. URL: <https://doi.org/10.3389/fbioe.2021.680257>.
- Elliott, D.M., D.A. Narmoneva, and L.A. Setton (2002). “Direct measurement of the poisson’s ratio of human patella cartilage in tension”. In: *Journal of Biomechanical Engineering* 124.2, pp. 223–228. ISSN: 0148-0731. URL: <https://doi.org/10.1115/1.1449905>.
- Erdemir, A. (2016). “Open Knee: Open source modeling and simulation in knee biomechanics”. EN. In: *J Knee Surg* 29.02, pp. 107–116. ISSN: 1538-8506. URL: <https://doi.org/10.1055/s-0035-1564600>.
- Eskelinen, A.S.A., M.E. Mononen, et al. (2019). “Maximum shear strain-based algorithm can predict proteoglycan loss in damaged articular cartilage”. In: *Biomechanics and Modeling in Mechanobiology* 18.3, pp. 753–778. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-018-01113-1>.
- Faisal, T.R., M. Adouni, and Y.Y. Dhaher (2023). “Surrogate modeling of articular cartilage degradation to understand the synergistic role of MMP-1 and MMP-9: A case study”. In: *Biomechanics and Modeling in Mechanobiology* 22.1, pp. 43–56. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-022-01630-0>.

- Fazekas, B. and T.J. Goda (2018). “Determination of the hyper-viscoelastic model parameters of open-cell polymer foams and rubber-like materials with high accuracy”. In: *Materials & Design* 156, pp. 596–608. ISSN: 0264-1275. URL: <https://doi.org/10.1016/j.matdes.2018.07.010>.
- Fehervary, H., L. Maes, J. Vastmans, et al. (2020). “How to implement user-defined fiber-reinforced hyperelastic materials in finite element software”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 110, p. 103737. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2020.103737>.
- Fernando, K.R.M. and C.P. Tsokos (2022). “Dynamically weighted balanced loss: Class imbalanced learning and confidence calibration of deep neural networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7, pp. 2940–2951. URL: <https://doi.org/10.1109/TNNLS.2020.3047335>.
- Fleisch, D.A. (2011). “A student’s guide to vectors and tensors”. In: URL: <https://doi.org/10.1017/CBO9781139031035>.
- Forrester, A.I.J., A. Sóbester, and A.J. Keane (2008). *Engineering Design via Surrogate Modelling*. Wiley. URL: <https://doi.org/10.1002/9780470770801>.
- Fortunato, M., T. Pfaff, P. Wirnsberger, et al. (2022). “MultiScale MeshGraphNets”. In: *ICML 2022 2nd AI for Science Workshop*. URL: <https://doi.org/10.48550/arXiv.2210.00612>.
- Frank, M., D. Drikakis, and V. Charassis (2020). “Machine-learning methods for computational science and engineering”. In: *Computation* 8.1. ISSN: 2079-3197. URL: <https://doi.org/10.3390/computation8010015>.
- Fresca, S. et al. (2021). “A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs”. In: *Journal of Scientific Computing* 87.2, p. 61. ISSN: 1573-7691. URL: <https://doi.org/10.1007/s10915-021-01462-7>.
- Fresca, S. and A. Manzoni (2022). “POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition”. In: *Computer Methods in Applied Mechanics and Engineering* 388, p. 114181. ISSN: 0045-7825. URL: <https://doi.org/10.1016/j.cma.2021.114181>.
- Fuks, O. and H.A. Tchelepi (2020). “Limitations of physics informed machine learning for nonlinear two-phase transport in porous media”. In: *Journal of Machine Learning for Modeling and Computing* 1.1, pp. 19–37. ISSN: 2689-3967.
- Fukushima, K. (1980). “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4, pp. 193–202. ISSN: 1432-0770. URL: <https://doi.org/10.1007/BF00344251>.
- Gao, F. and H. Zhong (2020). “Study on the large batch size training of neural networks based on the second order gradient”. In: *CoRR* abs/2012.08795. arXiv: [2012.08795](https://arxiv.org/abs/2012.08795). URL: <https://doi.org/10.48550/arXiv.2012.08795>.

- Gao, H. et al. (2021). “PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain”. In: *Journal of Computational Physics* 428, p. 110079. ISSN: 0021-9991. URL: <https://doi.org/10.1016/j.jcp.2020.110079>.
- Gao, W., X. Lu, et al. (2020). “A deep learning approach replacing the finite difference method for in situ stress prediction”. In: *IEEE Access* 8, pp. 44063–44074. ISSN: 2169-3536. URL: <https://doi.org/10.1109/ACCESS.2020.2977880>.
- Gee, M.W. et al. (2010). “A computational strategy for prestressing patient-specific biomechanical problems under finite deformation”. In: *International Journal for Numerical Methods in Biomedical Engineering* 26.1, pp. 52–72. URL: <https://doi.org/10.1002/cnm.1236>.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd. O'Reilly Media, Inc. ISBN: 9781492032649.
- Gibson, J., A. Hire, and R.G. Hennig (2022). “Data-augmentation for graph neural network learning of the relaxed energies of unrelaxed structures”. In: *npj Computational Materials* 8.1, p. 211. ISSN: 2057-3960. URL: <https://doi.org/10.1038/s41524-022-00891-8>.
- Giselle Fernández-Godino, M., S. Balachandar, and R.T. Haftka (2019). “On the use of symmetries in building surrogate models”. In: *Journal of Mechanical Design* 141.6. ISSN: 1050-0472. URL: <https://doi.org/10.1115/1.4042047>.
- Glorot, X. and Y. Bengio (2010). “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y.W. Teh and M. Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, pp. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- Goldblatt, J.P. and J.C. Richmond (2003). “Anatomy and biomechanics of the knee”. In: *Operative Techniques in Sports Medicine* 11.3. The Multiple Ligament Injured Knee, Part I, pp. 172–186. ISSN: 1060-1872. URL: <https://doi.org/10.1053/otsm.2003.35911>.
- Gómez-Florit, M., R.M.A. Domingues, S.M. Bakht, et al. (2020). “1.3.6 – Natural materials”. In: *Biomaterials Science*. Ed. by W.R. Wagner, S.E. Sakiyama-Elbert, G. Zhang, et al. 4th. Academic Press, pp. 361–375. ISBN: 978-0-12-816137-1. URL: <https://doi.org/10.1016/B978-0-12-816137-1.00026-X>.
- Govindjee, S. and P.A. Mihalic (1996). “Computational methods for inverse finite elastostatics”. In: *Computer Methods in Applied Mechanics and Engineering* 136.1, pp. 47–57. ISSN: 0045-7825. URL: [https://doi.org/10.1016/0045-7825\(96\)01045-6](https://doi.org/10.1016/0045-7825(96)01045-6).

- Guo, H. (2020). “Nonlinear Mixup: Out-of-manifold data augmentation for text classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04, pp. 4044–4051. URL: <https://doi.org/10.1609/aaai.v34i04.5822>.
- Guo, H. and Y. Mao (2023). “Interpolating graph pair to regularize graph classification”. In: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI’23/IAAI’23/EAAI’23. AAAI Press. ISBN: 978-1-57735-880-0. URL: <https://doi.org/10.1609/aaai.v37i6.25941>.
- Guo, H., Y. Mao, and R. Zhang (2019). “MixUp as locally linear out-of-manifold regularization”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, pp. 3714–3722. URL: <https://doi.org/10.1609/aaai.v33i01.33013714>.
- Hamsayeh Abbasi Niasar, E. and L.P. Li (2023). “Characterizing site-specific mechanical properties of knee cartilage with indentation-relaxation maps and machine learning”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 142, p. 105826. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2023.105826>.
- Han, X., Z. Jiang, et al. (2022). “G-Mixup: Graph Data Augmentation for Graph Classification”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 8230–8248. URL: <https://doi.org/10.48550/arXiv.2202.07179>.
- Harris, C.R., K.J. Millman, et al. (2020). “Array programming with NumPy”. In: *Nature* 585.7825, pp. 357–362. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- Haut Donahue, T.L., M.L. Hull, M.M. Rashid, et al. (2002). “A finite element model of the human knee joint for the study of tibio-femoral contact”. In: *Journal of Biomechanical Engineering* 124.3, pp. 273–280. ISSN: 0148-0731. DOI: [10.1115/1.1470171](https://doi.org/10.1115/1.1470171). URL: <https://doi.org/10.1115/1.1470171>.
- Hawker, G.A. (2019). “Osteoarthritis is a serious disease”. In: *Clinical and Experimental Rheumatology* 37 Suppl 120.5, pp. 3–6. ISSN: 0392-856X. URL: <http://europemc.org/abstract/med/31621562>.
- He, k. et al. (2015). “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. URL: <https://dx.doi.org/10.1109/ICCV.2015.123>.
- Hill, T.L. (1956). “A. Fundamental studies. On the theory of the Donnan membrane equilibrium”. In: *Discuss. Faraday Soc.* 21 (0), pp. 31–45. URL: <http://dx.doi.org/10.1039/DF9562100031>.
- Hinton, G., N. Srivastava, and K. Swersky (2012). *Neural Networks for Machine Learning*. Coursera Slide Presentation. Lecture 6a: Overview of mini-batch gradient descent.

URL: [https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf).

- Hinton, G.E. and R.R. Salakhutdinov (2006). “Reducing the dimensionality of data with neural networks”. In: *Science* 313.5786, pp. 504–507. URL: <https://doi.org/10.1126/science.1127647>.
- Hintze, J.L. and R.D. Nelson (1998). “Violin plots: A box plot-density trace synergism”. In: *The American Statistician* 52.2, pp. 181–184. URL: <https://doi.org/10.1080/00031305.1998.10480559>.
- Hoerig, C., J. Ghaboussi, and F. Insana M. (2020). “Physics-guided machine learning for 3-D quantitative quasi-static elasticity imaging”. In: *Physics in Medicine & Biology* 65.6, p. 065011. URL: <https://dx.doi.org/10.1088/1361-6560/ab7505>.
- Hoffer, J.G., A.B. Ofner, F.M. Rohrhofer, et al. (2022). “Theory-inspired machine learning—towards a synergy between knowledge and data”. In: *Welding in the World* 66.7, pp. 1291–1304. ISSN: 1878-6669. URL: <https://doi.org/10.1007/s40194-022-01270-z>.
- Holzapfel, G.A. (2000). *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Chichester: John Wiley & Sons.
- Holzapfel, G.A., K. Linka, et al. (2021). “Predictive constitutive modelling of arteries by deep learning”. In: *Journal of The Royal Society Interface* 18.182, p. 20210411. URL: <https://doi.org/10.1098/rsif.2021.0411>.
- Hosseini, S.M., W. Wilson, et al. (2014). “A numerical model to study mechanically induced initiation and progression of damage in articular cartilage”. In: *Osteoarthritis and Cartilage* 22.1, pp. 95–103. ISSN: 1063-4584. URL: <https://doi.org/10.1016/j.joca.2013.10.010>.
- Hunter, J.D. (2007). “Matplotlib: A 2D graphics environment”. In: *Computing in Science and Engineering* 9.3, pp. 90–95. ISSN: 15219615. URL: <https://doi.org/10.1109/MCSE.2007.55>.
- Ioffe, S. and C. Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456. URL: <https://doi.org/10.48550/arXiv.1502.03167>.
- Jahya, A., M. Herink, and S. Misra (2013). “A framework for predicting three-dimensional prostate deformation in real time”. In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 9.4, e52–e60. URL: <https://doi.org/10.1002/rcs.1493>.
- Jindal, A., A. Ghosh Chowdhury, A. Didolkar, et al. (2020). “Augmenting NLP models using latent feature interpolations”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by D. Scott, N. Bel, and C. Zong. Barcelona, Spain

- (Online): International Committee on Computational Linguistics, pp. 6931–6936. URL: <https://doi.org/10.18653/v1/2020.coling-main.611>.
- Johnsen, S.F., Z.A. Taylor, M.J. Clarkson, et al. (2015). “NiftySim: A GPU-based nonlinear finite element package for simulation of soft tissue biomechanics”. In: *International Journal of Computer Assisted Radiology and Surgery* 10.7, pp. 1077–1095. ISSN: 1861-6429. URL: <https://doi.org/10.1007/s11548-014-1118-5>.
- Johnson, D., R. Hashaikeh, and N. Hilal (2021). “1 - Basic principles of osmosis and osmotic pressure”. In: *Osmosis Engineering*. Ed. by N. Hilal, A.F. Ismail, M. Khayet, et al. Elsevier, pp. 1–15. ISBN: 978-0-12-821016-1. URL: <https://doi.org/10.1016/B978-0-12-821016-1.00011-5>.
- Julkunen, P., W. Wilson, J.S. Jurvelin, et al. (2008). “Stress-relaxation of human patellar articular cartilage in unconfined compression: Prediction of mechanical response by tissue composition and structure”. In: *Journal of Biomechanics* 41.9, pp. 1978–1986. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2008.03.026>.
- K., Thomas et al. (2016). “Jupyter Notebooks – a publishing format for reproducible computational workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. IOS Press, pp. 87–90. ISBN: 9781614996484. URL: <http://dx.doi.org/10.3233/978-1-61499-649-1-87>.
- Kalyuzhnyuk, A.V., R.L. Lapin, A.S. Murachev, et al. (2019). “Neural networks and data-driven surrogate models for simulation of steady-state fracture growth”. In: *Materials Physics and Mechanics* 42.3, pp. 351–358. ISSN: 16058119. URL: [http://dx.doi.org/10.18720/MPM.4232019\\_10](http://dx.doi.org/10.18720/MPM.4232019_10).
- Karniadakis, G.E., I.G. Kevrekidis, L. Lu, et al. (2021). “Physics-informed machine learning”. In: *Nature Reviews Physics* 3.6, pp. 422–440. ISSN: 2522-5820. URL: <https://doi.org/10.1038/s42254-021-00314-5>.
- Kashinath, K., M. Mustafa, A. Albert, et al. (2021). “Physics-informed machine learning: case studies for weather and climate modelling”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194, p. 20200093. URL: <https://doi.org/10.1098/rsta.2020.0093>.
- Kazemi, M., L.P. Li, P. Savard, et al. (2011). “Creep behavior of the intact and meniscectomy knee joints”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 4.7, pp. 1351–1358. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2011.05.004>.
- Keenan, K.E., S. Pal, D.P. Lindsey, et al. (2013). “A viscoelastic constitutive model can accurately represent entire creep indentation tests of human patella cartilage”. In: *Journal of Applied Biomechanics* 29.3, pp. 292–302. URL: <https://doi.org/10.1123/jab.29.3.292>.

- Khaniki, H.B., M.H. Ghayesh, et al. (2023). "Hyperelastic structures: A review on the mechanics and biomechanics". In: *International Journal of Non-Linear Mechanics* 148, p. 104275. ISSN: 0020-7462. URL: <https://doi.org/10.1016/j.ijnonlinmec.2022.104275>.
- Kiani, C., L. Chen, Y.J. Wu, et al. (2002). "Structure and function of aggrecan". In: *Cell Research* 12.1, pp. 19–32. ISSN: 1748-7838. URL: <https://doi.org/10.1038/sj.cr.7290106>.
- Kiefer, J. and J. Wolfowitz (1952). "Stochastic estimation of the maximum of a regression function". In: *The Annals of Mathematical Statistics* 23.3, pp. 462–466. ISSN: 00034851. URL: <http://www.jstor.org/stable/2236690> (visited on 12/19/2023).
- Kim, B., S.B. Lee, J. Lee, et al. (2012). "A comparison among neo-Hookean model, Mooney-Rivlin model, and Ogden model for chloroprene rubber". In: *International Journal of Precision Engineering and Manufacturing* 13.5, pp. 759–764. ISSN: 2005-4602. URL: <https://doi.org/10.1007/s12541-012-0099-y>.
- Kingma, D.P., S. Mohamed, et al. (2014). "Semi-supervised learning with deep generative models". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, et al. Vol. 27. Curran Associates, Inc. URL: <https://doi.org/10.48550/arXiv.1406.5298>.
- Klambauer, G. et al. (2017). "Self-normalizing neural networks". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: <https://doi.org/10.48550/arXiv.1706.02515>.
- Klika, V., E.A. Gaffney, Y.C. Chen, et al. (2016). "An overview of multiphase cartilage mechanical modelling and its role in understanding function and pathology". In: *Journal of the Mechanical Behavior of Biomedical Materials* 62, pp. 139–157. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2016.04.032>.
- Klisch, S.M. (1999). "Internally constrained mixtures of elastic continua". In: *Mathematics and Mechanics of Solids* 4.4, pp. 481–498. URL: <https://doi.org/10.1177/108128659900400405>.
- Kong, H., X.Q. Wang, and X.A. Zhang (2022). "Exercise for osteoarthritis: A literature review of pathology and mechanism". In: *Frontiers in Aging Neuroscience* 14. ISSN: 1663-4365. URL: <https://doi.org/10.3389/fnagi.2022.854026>.
- Lai, W.M., J.S. Hou, and V.C. Mow (1991). "A triphasic theory for the swelling and deformation behaviors of articular cartilage". In: *Journal of Biomechanical Engineering* 113.3, pp. 245–258. ISSN: 0148-0731. URL: <https://doi.org/10.1115/1.2894880>.
- Lauzeral, N., D. Borzacchiello, M. Kugler, et al. (2019). "A model order reduction approach to create patient-specific mechanical models of human liver in computational medicine applications". In: *Computer Methods and Programs in Biomedicine* 170, pp. 95–106. ISSN: 0169-2607. URL: <https://doi.org/10.1016/j.cmpb.2019.01.003>.

- Leach, J.R., C. Zhu, et al. (2019). “Comparison of two methods for estimating the unloaded state for abdominal aortic aneurysm stress calculations”. In: *Journal of Mechanics in Medicine and Biology* 19.03, p. 1950015. URL: <https://doi.org/10.1142/S0219519419500155>.
- LeCun, Y.A., L. Bottou, et al. (2012). “Efficient BackProp”. In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by G. Montavon et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 9–48. ISBN: 978-3-642-35289-8. URL: [https://doi.org/10.1007/978-3-642-35289-8\\_3](https://doi.org/10.1007/978-3-642-35289-8_3).
- Lenhart, R.L., J. Kaiser, et al. (2015). “Prediction and validation of load-dependent behavior of the tibiofemoral and patellofemoral joints during movement”. In: *Annals of Biomedical Engineering* 43.11, pp. 2675–2685. ISSN: 1573-9686. URL: <https://doi.org/10.1007/s10439-015-1326-3>.
- Lespasio, M.J., N.S. Piuzzi, M.E. Husni, et al. (2017). “Knee osteoarthritis: A primer”. In: *The Permanente Journal* 21, pp. 16–183. ISSN: 1552-5767. URL: <https://doi.org/10.7812/tpp/16-183>.
- Li, Z., N. Kovachki, K. Azizzadenesheli, et al. (2020). “Multipole graph neural operator for parametric partial differential equations”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. Vol. 33. Curran Associates, Inc., pp. 6755–6766. URL: <https://doi.org/10.48550/arXiv.2006.09535>.
- Liang, L., M. Liu, et al. (2018). “A machine learning approach as a surrogate of finite element analysis-based inverse method to estimate the zero-pressure geometry of human thoracic aorta”. In: *International Journal for Numerical Methods in Biomedical Engineering* 34.8, e3103. ISSN: 20407939. URL: <https://doi.org/10.1002/cnm.3103>.
- Lin, W., Q. Meng, J. Li, et al. (2021). “The effect of highly inhomogeneous biphasic properties on mechanical behaviour of articular cartilage”. In: *Computer Methods and Programs in Biomedicine* 206, p. 106122. ISSN: 0169-2607. URL: <https://doi.org/10.1016/j.cmpb.2021.106122>.
- Linka, K. et al. (2021). “Constitutive artificial neural networks: A fast and general approach to predictive data-driven constitutive modeling by deep learning”. In: *Journal of Computational Physics* 429, p. 110010. ISSN: 0021-9991. URL: <https://doi.org/10.1016/j.jcp.2020.110010>.
- Lipshitz, H. et al. (1975). “In vitro wear of articular cartilage”. In: *JBJS* 57.4. ISSN: 0021-9355. URL: [https://journals.lww.com/jbjsjournal/fulltext/1975/57040/in\\_vitro\\_wear\\_of\\_articular\\_cartilage.15.aspx](https://journals.lww.com/jbjsjournal/fulltext/1975/57040/in_vitro_wear_of_articular_cartilage.15.aspx).
- Liu, D., S. Ma, et al. (2020). “A biphasic visco-hyperelastic damage model for articular cartilage: application to micromechanical modelling of the osteoarthritis-induced degradation behaviour”. In: *Biomechanics and Modeling in Mechanobiology* 19.3, pp. 1055–1077. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-019-01270-x>.

- Liu, G.R. (2019). “FEA-AI and AI-AI: Two-way Deepnets for real-time computations for both forward and inverse mechanics problems”. In: *International Journal of Computational Methods* 16.08, p. 1950045. URL: <https://doi.org/10.1142/S0219876219500452>.
- Loeser, R.F., J.A. Collins, and B.O. Diekman (2016). “Ageing and the pathogenesis of osteoarthritis”. In: *Nature Reviews Rheumatology* 12.7, pp. 412–420. ISSN: 1759-4804. URL: <https://doi.org/10.1038/nrrheum.2016.65>.
- Lostado Lorza, R., F. Somovilla Gomez, M. Corral Bobadilla, et al. (2021). “Comparative analysis of healthy and cam-type femoroacetabular impingement (FAI) human hip joints using the finite element method”. In: *Applied Sciences* 11.23. ISSN: 2076-3417. URL: <https://doi.org/10.3390/app112311101>.
- Lu, X.L. and V.C. Mow (2008). “Biomechanics of articular cartilage and determination of material properties”. In: *Medicine & Science in Sports & Exercise* 40.2. ISSN: 0195-9131. URL: <https://doi.org/10.1249/mss.0b013e31815cb1fc>.
- Maes, L. and N. Famaey (2023). “How to implement constrained mixture growth and remodeling algorithms for soft biological tissues”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 140, p. 105733. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2023.105733>.
- Maes, L., H. Fehervary, J. Vastmans, et al. (2019). “Constrained mixture modeling affects material parameter identification from planar biaxial tests”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 95, pp. 124–135. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2019.03.029>.
- Mameri, E.S., S.P. Dasari, L.M. Fortier, et al. (2022). “Review of meniscus anatomy and biomechanics”. In: *Current Reviews in Musculoskeletal Medicine* 15.5, pp. 323–335. ISSN: 1935-9748. URL: <https://doi.org/10.1007/s12178-022-09768-1>.
- Maneeuwongvatana, S. and D.M. Mount (1999). “Analysis of approximate nearest neighbor searching with clustered point sets”. In: *CoRR cs.CG/9901013*. URL: <https://doi.org/10.48550/arXiv.cs/9901013>.
- Marinkovic, D. and M. Zehn (2019). “Survey of finite element method-based real-time simulations”. In: *Applied Sciences* 9.14. ISSN: 2076-3417. URL: <https://doi.org/10.3390/app9142775>.
- Martin-Guerrero, J.D., M.J. Ruperez-Moreno, F. Martinez-Martinez, et al. (2016). “Machine learning for modeling the biomechanical behavior of human soft tissue”. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 247–253. URL: <https://doi.org/10.1109/ICDMW.2016.0042>.
- Martínez-Martínez, F., M.J. Rupérez-Moreno, et al. (2017). “A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time”. In: *Computers in Biology and Medicine* 90, pp. 116–124. ISSN: 0010-4825. URL: <https://doi.org/10.1016/j.combiomed.2017.09.019>.

- Martínez-Moreno, D., G. Jiménez, P. Gálvez-Martín, et al. (2019). “Cartilage biomechanics: A key factor for osteoarthritis regenerative medicine”. In: *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease* 1865.6, pp. 1067–1075. ISSN: 0925-4439. URL: <https://doi.org/10.1016/j.bbadi.2019.03.011>.
- Miehe, C. (1996). “Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity”. In: *Computer Methods in Applied Mechanics and Engineering* 134.3, pp. 223–240. ISSN: 0045-7825. URL: [https://doi.org/10.1016/0045-7825\(96\)01019-5](https://doi.org/10.1016/0045-7825(96)01019-5).
- Mirmajarabian, S.A., A.W. Kajabi, J.H.J. Ketola, et al. (2023). “Machine learning prediction of collagen fiber orientation and proteoglycan content from multiparametric quantitative MRI in articular cartilage”. In: *Journal of Magnetic Resonance Imaging* 57.4, pp. 1056–1068. URL: <https://doi.org/10.1002/jmri.28353>.
- Mobasher, A., M.P. Rayman, O. Gualillo, et al. (2017). “The role of metabolism in the pathogenesis of osteoarthritis”. In: *Nature Reviews Rheumatology* 13.5, pp. 302–311. ISSN: 1759-4804. URL: <https://doi.org/10.1038/nrrheum.2017.50>.
- Molnar, C. (2019). *Interpretable machine learning. A Guide for Making Black Box Models Explainable*. URL: <https://christophm.github.io/interpretable-ml-book>.
- Mononen, M.E., M.T. Mikkola, P. Julkunen, et al. (2012). “Effect of superficial collagen patterns and fibrillation of femoral articular cartilage on knee joint mechanics—A 3D finite element analysis”. In: *Journal of Biomechanics* 45.3, pp. 579–587. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2011.11.003>.
- Mononen, M.E., A. Paz, et al. (2023). “Atlas-based finite element analyses with simpler constitutive models predict personalized progression of knee osteoarthritis: data from the osteoarthritis initiative”. In: *Scientific Reports* 13.1, p. 8888. ISSN: 2045-2322. URL: <https://doi.org/10.1038/s41598-023-35832-y>.
- Motavalli, M., O. Akkus, and J.M. Mansour (2014). “Depth-dependent shear behavior of bovine articular cartilage: Relationship to structure”. In: *Journal of Anatomy* 225.5, pp. 519–526. URL: <https://doi.org/10.1111/joa.12230>.
- Mow, V.C., S.C. Kuei, W.M. Lai, et al. (1980). “Biphasic creep and stress relaxation of articular cartilage in compression: Theory and experiments”. In: *Journal of Biomechanical Engineering* 102.1, pp. 73–84. ISSN: 0148-0731. DOI: [10.1115/1.3138202](https://doi.org/10.1115/1.3138202). URL: <https://doi.org/10.1115/1.3138202>.
- Mumuni, A. and F. Mumuni (2022). “Data augmentation: A comprehensive survey of modern approaches”. In: *Array* 16, p. 100258. ISSN: 2590-0056. URL: <https://doi.org/10.1016/j.array.2022.100258>.
- Naghibi Beidokhti, H., D. Janssen, M. Khoshgoftar, et al. (2016). “A comparison between dynamic implicit and explicit finite element simulations of the native knee joint”. In: *Medical Engineering & Physics* 38.10, pp. 1123–1130. ISSN: 1350-4533. URL: <https://doi.org/10.1016/j.medengphy.2016.06.001>.

- Nair, V. and G.E. Hinton (2010). “Rectified linear units improve restricted Boltzmann machines”. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML’10, pp. 807–814. ISBN: 9781605589077. URL: <https://icml.cc/Conferences/2010/papers/432.pdf>.
- Narmoneva, D.A., J.Y. Wang, and L.A. Setton (1999). “Nonuniform swelling-induced residual strains in articular cartilage”. In: *Journal of Biomechanics* 32.4, pp. 401–408. ISSN: 0021-9290. URL: [https://doi.org/10.1016/S0021-9290\(98\)00184-5](https://doi.org/10.1016/S0021-9290(98)00184-5).
- Nesterov, Y. (1983). “A method for solving the convex programming problem with convergence rate  $O(1/k^2)$ ”. In: *Proceedings of the USSR Academy of Sciences* 269, pp. 543–547. URL: <https://api.semanticscholar.org/CorpusID:145918791>.
- Niroomandi, S., I. Alfaro, et al. (2012). “Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models”. In: *Computer Methods and Programs in Biomedicine* 105.1, pp. 1–12. ISSN: 0169-2607. URL: <https://doi.org/10.1016/j.cmpb.2010.06.012>.
- Nolan, D.R., C. Lally, and J.P. McGarry (2022). “Understanding the deformation gradient in Abaqus and key guidelines for anisotropic hyperelastic user material subroutines (UMATs)”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 126, p. 104940. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2021.104940>.
- Nourbakhsh, M., J. Irizarry, and J. Haymaker (2018). “Generalizable surrogate model features to approximate stress in 3D trusses”. In: *Engineering Applications of Artificial Intelligence* 71, pp. 15–27. ISSN: 0952-1976. URL: <https://doi.org/10.1016/j.engappai.2018.01.006>.
- Occhetta, P., A. Mainardi, E. Votta, et al. (2019). “Hyperphysiological compression of articular cartilage induces an osteoarthritic phenotype in a cartilage-on-a-chip model”. In: *Nature Biomedical Engineering* 3.7, pp. 545–557. ISSN: 2157-846X. URL: <https://doi.org/10.1038/s41551-019-0406-3>.
- Oh, C., S. Han, and J. Jeong (2020). “Time-series data augmentation based on interpolation”. In: *Procedia Computer Science* 175. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology, pp. 64–71. ISSN: 1877-0509. URL: <https://doi.org/10.1016/j.procs.2020.07.012>.
- Orava, H., L. Huang, S.P. Ojanen, et al. (2022). “Changes in subchondral bone structure and mechanical properties do not substantially affect cartilage mechanical responses – A finite element study”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 128, p. 105129. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2022.105129>.
- Orozco, G.A., P. Tanska, A. Gustafsson, et al. (2022). “Crack propagation in articular cartilage under cyclic loading using cohesive finite element modeling”. In: *Journal of*

- the Mechanical Behavior of Biomedical Materials* 131, p. 105227. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2022.105227>.
- Paiva, G., S. Bhashyam, G. Thiagarajan, et al. (2012). “A data-driven surrogate model to connect scales between multi-domain biomechanics simulations”. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3077–3080. URL: <https://doi.org/10.1109/EMBC.2012.6346614>.
- Pan, S.J., I.W. Tsang, et al. (2011). “Domain adaptation via transfer component analysis”. In: *IEEE Transactions on Neural Networks* 22.2, pp. 199–210. URL: <https://doi.org/10.1109/TNN.2010.2091281>.
- Pandolfi, A. and G.A. Holzapfel (2008). “Three-dimensional modeling and computational analysis of the human cornea considering distributed collagen fibril orientations”. In: *Journal of Biomechanical Engineering* 130.6, p. 061006. ISSN: 0148-0731. URL: <https://doi.org/10.1115/1.2982251>.
- Panfilov, E., A. Tiulpin, S. Klein, et al. (2019). “Improving robustness of deep learning based knee MRI segmentation: Mixup and adversarial domain adaptation”. In: *The IEEE International Conference on Computer Vision (ICCV) Workshops*. URL: <https://doi.org/10.48550/arXiv.1908.04126>.
- Panfilov, E., A. Tiulpin, M.T. Nieminen, et al. (2022). “Deep learning-based segmentation of knee MRI for fully automatic subregional morphological assessment of cartilage tissues: Data from the Osteoarthritis Initiative”. In: *Journal of Orthopaedic Research* 40.5, pp. 1113–1124. URL: <https://doi.org/10.1002/jor.25150>.
- Pant, P., R. Doshi, et al. (2021). “Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations”. In: *Physics of Fluids* 33.10, p. 107101. ISSN: 1070-6631. URL: <https://doi.org/10.1063/5.0062546>.
- Parida, S.S., S. Bose, and G. Apostolakis (2023). “Earthquake data augmentation using wavelet transform for training deep learning based surrogate models of nonlinear structures”. In: *Structures* 55, pp. 638–649. ISSN: 2352-0124. URL: <https://doi.org/10.1016/j.istruc.2023.05.122>.
- Park, J., H. Shim, and E. Yang (2022). “Graph transplant: Node saliency-guided graph mixup with local structure preservation”. In: *Proceedings of the First MiniCon Conference*. URL: <https://doi.org/10.48550/arXiv.2111.05639>.
- Paz, A., G.A. Orozco, P. Tanska, et al. (2022). “A novel knee joint model in FEBio with inhomogeneous fibril-reinforced biphasic cartilage simulating tissue mechanical responses during gait: Data from the osteoarthritis initiative”. In: *Computer Methods in Biomechanics and Biomedical Engineering* 0.0, pp. 1–15. URL: <https://doi.org/10.1080/10255842.2022.2117548>.
- Pearle, A.D., R.F. Warren, and S.A. Rodeo (2005). “Basic science of articular cartilage and osteoarthritis”. In: *Clinics in Sports Medicine* 24.1, pp. 1–12. ISSN: 0278-5919. URL: <https://doi.org/10.1016/j.csm.2004.08.007>.

- Pedregosa, F., G. Varoquaux, A. Gramfort, et al. (2011). “Scikit-learn: Machine learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830. URL: <https://doi.org/10.48550/arXiv.1201.0490>.
- Pehnerstorfer, B., K. Willcox, and M. Gunzburger (2018). “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization”. In: *SIAM Review* 60.3, pp. 550–591. URL: <https://doi.org/10.1137/16M1082469>.
- Pellicer-Valero, O.J., M.J. Rupérez, et al. (2020). “Real-time biomechanical modeling of the liver using Machine Learning models trained on Finite Element Method simulations”. In: *Expert Systems with Applications* 143, p. 113083. ISSN: 0957-4174. URL: <https://doi.org/10.1016/j.eswa.2019.113083>.
- Pfaff, T., M. Fortunato, et al. (2021). “Learning mesh-based simulation with graph networks”. In: *International Conference on Learning Representations*, pp. 1–18. ISSN: 23318422. URL: <https://doi.org/10.48550/arXiv.2010.03409>.
- Phellan, R., B. Hachem, J. Clin, et al. (2021). “Real-time biomechanics using the finite element method and machine learning: Review and perspective”. In: *Medical Physics* 48.1, pp. 7–18. ISSN: 00942405. URL: <https://doi.org/10.1002/mp.14602>.
- Pierce, D.M., T.E. Fastl, et al. (2015). “A method for incorporating three-dimensional residual stretches/stresses into patient-specific finite element simulations of arteries”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 47, pp. 147–164. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2015.03.024>.
- Polyak, B.T. (1964). “Some methods of speeding up the convergence of iteration methods”. In: *USSR Computational Mathematics and Mathematical Physics* 4.5, pp. 1–17. ISSN: 0041-5553. URL: [https://doi.org/10.1016/0041-5553\(64\)90137-5](https://doi.org/10.1016/0041-5553(64)90137-5).
- Pore, T., S.G. Thorat, and A.A. Nema (2021). “Review of contact modelling in nonlinear finite element analysis”. In: *Materials Today: Proceedings* 47. International Conference on Materials and System Engineering, pp. 2436–2440. ISSN: 2214-7853. URL: <https://doi.org/10.1016/j.matpr.2021.04.504>.
- Quiroga, J.M.P., W. Wilson, K. Ito, et al. (2017). “Relative contribution of articular cartilage’s constitutive components to load support depending on strain rate”. In: *Biomechanics and Modeling in Mechanobiology* 16.1, pp. 151–158. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-016-0807-0>.
- Rachev, A. and S.E. Greenwald (2003). “Residual strains in conduit arteries”. In: *Journal of Biomechanics* 36.5. Cardiovascular Biomechanics, pp. 661–670. ISSN: 0021-9290. URL: [https://doi.org/10.1016/S0021-9290\(02\)00444-X](https://doi.org/10.1016/S0021-9290(02)00444-X).
- Rauschenberger, M. and R. Baeza-Yates (2020). “How to handle health-related small imbalanced data in machine learning?” In: *i-com* 19.3, pp. 215–226. URL: <https://doi.org/10.1515/icom-2020-0018>.
- Reynolds, M., J. Rae, A. Fidjeland, et al. (2017). *Open sourcing Sonnet - a new library for constructing neural networks*. <https://www.deepmind.com/blog/open->

[sourcing - sonnet - a - new - library - for - constructing - neural - networks](#) (retrieved on 28/04/22).

- Rezende, D.J., S. Mohamed, and D. Wierstra (2014). “Stochastic backpropagation and approximate inference in deep generative models”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E.P. Xing and T. Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1278–1286. URL: <https://doi.org/10.48550/arXiv.1401.4082>.
- Rezvani, S. and X. Wang (2023). “A broad review on class imbalance learning techniques”. In: *Applied Soft Computing* 143, p. 110415. ISSN: 1568-4946. URL: <https://doi.org/10.1016/j.asoc.2023.110415>.
- Rieppo, J., M.M. Hyttinen, R. Lappalainen, et al. (2004). “Spatial determination of water, collagen and proteoglycan contents by Fourier transform infrared imaging and digital densitometry”. In: *50th Annual Meeting of the Orthopaedic Research Society*.
- Rumelhart, D.E., G.E. Hinton, and R.J. Williams (1986). “Learning representations by back-propagating errors”. In: *Nature* 323.6088, pp. 533–536. ISSN: 1476-4687. URL: <https://doi.org/10.1038/323533a0>.
- Sajjadinia, S.S. and M. Haghpanahi (2021). “A parametric study on the mechanical role of fibrillar rotations in an articular cartilage finite element model”. In: *Scientia Iranica* 28.2, pp. 830–836. ISSN: 1026-3098. URL: <https://dx.doi.org/10.24200/sci.2020.51785.2362>.
- Sajjadinia, S.S., M. Haghpanahi, and M. Razi (2019). “Computational simulation of the multiphasic degeneration of the bone-cartilage unit during osteoarthritis via indentation and unconfined compression tests”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 233.9, pp. 871–882. URL: <https://doi.org/10.1177/0954411919854011>.
- Salmon, J.H., A.C. Rat, J. Sellam, et al. (2016). “Economic impact of lower-limb osteoarthritis worldwide: A systematic review of cost-of-illness studies”. In: *Osteoarthritis and Cartilage* 24.9, pp. 1500–1508. ISSN: 1063-4584. URL: <https://doi.org/10.1016/j.joca.2016.03.012>.
- Sanchez-Gonzalez, A., J. Godwin, T. Pfaff, et al. (2020). “Learning to simulate complex physics with graph networks”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 8459–8468. URL: <https://doi.org/10.48550/arXiv.2002.09405>.
- Sella, V., J. Pham, et al. (2023). “Projection-based multifidelity linear regression for data-poor applications”. In: *AIAA SCITECH 2023 Forum*. URL: <https://doi.org/10.2514/6.2023-0916>.
- Setton, L.A., W. Gu, et al. (1996). “Predictions of the swelling-induced pre-stress in articular cartilage”. In: *Mechanics of Poroelastic Media*. Ed. by A.P.S. Selvadurai.

- Dordrecht: Springer Netherlands, pp. 299–320. ISBN: 978-94-015-8698-6. URL: [https://doi.org/10.1007/978-94-015-8698-6\\_17](https://doi.org/10.1007/978-94-015-8698-6_17).
- Shalev-Shwartz, S. and Y. Wexler (2016). “Minimizing the Maximal Loss: How and Why”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 793–801. URL: <https://proceedings.mlr.press/v48/shalev-shwartzb16.html>.
- Shapiro, E.M., A. Borthakur, J.H. Kaufman, et al. (2001). “Water distribution patterns inside bovine articular cartilage as visualized by <sup>1</sup>H magnetic resonance imaging”. In: *Osteoarthritis and Cartilage* 9.6, pp. 533–538. ISSN: 1063-4584. URL: <https://doi.org/10.1053/joca.2001.0428>.
- Shim, V.B., T.F. Besier, D.G. Lloyd, et al. (2016). “The influence and biomechanical role of cartilage split line pattern on tibiofemoral cartilage stress distribution during the stance phase of gait”. In: *Biomechanics and Modeling in Mechanobiology* 15.1, pp. 195–204. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-015-0668-y>.
- Shim, V.B., S. Holdsworth, A.A. Champagne, et al. (2020). “Rapid prediction of brain injury pattern in mTBI by combining FE analysis With a machine-learning based approach”. In: *IEEE Access* 8, pp. 179457–179465. URL: <https://doi.org/10.1109/ACCESS.2020.3026350>.
- Shirazi, R. and A. Shirazi-Adl (2008). “Deep vertical collagen fibrils play a significant role in mechanics of articular cartilage”. In: *Journal of Orthopaedic Research* 26.5, pp. 608–615. URL: <https://doi.org/10.1002/jor.20537>.
- Simo, J.C. (1987). “On a fully three-dimensional finite-strain viscoelastic damage model: Formulation and computational aspects”. In: *Computer Methods in Applied Mechanics and Engineering* 60.2, pp. 153–173. ISSN: 0045-7825. URL: [https://doi.org/10.1016/0045-7825\(87\)90107-1](https://doi.org/10.1016/0045-7825(87)90107-1).
- Singh, D. and B. Singh (2022). “Feature wise normalization: An effective way of normalizing data”. In: *Pattern Recognition* 122, p. 108307. ISSN: 0031-3203. URL: <https://doi.org/10.1016/j.patcog.2021.108307>.
- Steininger, M., K. Kobs, P. Davidson, et al. (2021). “Density-based weighting for imbalanced regression”. In: *Machine Learning* 110.8, pp. 2187–2211. ISSN: 1573-0565. URL: <https://doi.org/10.1007/s10994-021-06023-5>.
- Stender, M.E., C.B. Raub, K.A. Yamauchi, et al. (2013). “Integrating qPLM and biomechanical test data with an anisotropic fiber distribution model and predictions of TGF- $\beta$ 1 and IGF-1 regulation of articular cartilage fiber modulus”. In: *Biomechanics and Modeling in Mechanobiology* 12.6, pp. 1073–1088. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-012-0463-y>.
- Stender, M.E., R.A. Regueiro, and V.L. Ferguson (2017). “A poroelastic finite element model of the bone–cartilage unit to determine the effects of changes in permeability

- with osteoarthritis". In: *Computer Methods in Biomechanics and Biomedical Engineering* 20.3, pp. 319–331. URL: <https://doi.org/10.1080/10255842.2016.1233326>.
- Strang, G. (2023). *Introduction to linear algebra*. 6th. Cambridge University Press. URL: <https://math.mit.edu/~gs/linearalgebra/ila6/indexila6.html>.
- Sun, F., J. Hoffmann, et al. (2020). "InfoGraph: Unsupervised and Semi-supervised Graph-Level Representation Learning via Mutual Information Maximization". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia*. URL: <https://doi.org/10.48550/arXiv.1908.01000>.
- Taber, L.A. and D.W. Eggers (1996). "Theoretical study of stress-modulated growth in the aorta". In: *Journal of Theoretical Biology* 180.4, pp. 343–357. ISSN: 0022-5193. URL: <https://doi.org/10.1006/jtbi.1996.0107>.
- Taghizadeh, S., F.D. Witherden, and S.S. Girimaji (2020). "Turbulence closure modeling with data-driven techniques: physical compatibility and consistency considerations". In: *New Journal of Physics* 22.9, p. 093023. URL: <https://dx.doi.org/10.1088/1367-2630/abadb3>.
- Tan, M. and Q.V. Le (2021). "EfficientNetV2: Smaller models and faster training". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 10096–10106. URL: <https://doi.org/10.48550/arXiv.2104.00298>.
- Terzaghi, K. (1943). *Theoretical Soil Mechanics*. Hoboken, NJ, USA: John Wiley & Sons, Inc. URL: <https://doi.org/10.1002/9780470172766>.
- Thienkarochanakul, K., A.A. Javadi, M. Akrami, et al. (2020). "Stress distribution of the tibiofemoral joint in a healthy versus osteoarthritis knee model using image-based three-dimensional finite element analysis". In: *Journal of Medical and Biological Engineering* 40.3, pp. 409–418. ISSN: 2199-4757. URL: <https://doi.org/10.1007/s40846-020-00523-w>.
- Thomas, K.A., D. Krzemiński, L. Kidziński, et al. (2021). "Open source software for automatic subregional assessment of knee cartilage degradation using quantitative T2 relaxometry and deep learning". In: *Cartilage* 13.1\_suppl, 747S–756S. URL: <https://doi.org/10.1177/19476035211042406>.
- Um, K., R. Brand, et al. (2020). "Solver-in-the-loop: learning from differentiable physics to interact with iterative PDE-solvers". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc. ISBN: 9781713829546. URL: <https://doi.org/10.48550/arXiv.2007.00016>.
- Veličković, P., W. Fedus, W.L. Hamilton, et al. (2019). "Deep Graph Infomax". In: *International Conference on Learning Representations*. URL: <https://doi.org/10.48550/arXiv.1809.10341>.

- Verruijt, A. (1984). “The theory of consolidation”. In: *Fundamentals of Transport Phenomena in Porous Media*. Ed. by J. Bear and M.Y. Corapcioglu. Dordrecht: Springer Netherlands, pp. 349–368. ISBN: 978-94-009-6175-3. URL: [https://doi.org/10.1007/978-94-009-6175-3\\_7](https://doi.org/10.1007/978-94-009-6175-3_7).
- Vincent, P., H. Larochelle, et al. (2008). “Extracting and composing robust features with denoising autoencoders”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, pp. 1096–1103. ISBN: 9781605582054. URL: <https://doi.org/10.1145/1390156.1390294>.
- Virtanen, P., R. Gommers, T.E. Oliphant, et al. (2020). “SciPy 1.0: Fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17, pp. 261–272. URL: <https://doi.org/10.1038/s41592-019-0686-2>.
- Votta, E., M. Presicce, A. Della Corte, et al. (2017). “A novel approach to the quantification of aortic root in vivo structural mechanics”. In: *International Journal for Numerical Methods in Biomedical Engineering* 33.9, e2849. URL: <https://doi.org/10.1002/cnm.2849>.
- Walker, P.S. and M.J. Erkman (1975). “The role of the menisci in force transmission across the knee”. In: *Clinical Orthopaedics and Related Research* 109, pp. 184–192. ISSN: 0009-921X. URL: <https://doi.org/10.1097/00003086-197506000-00027>.
- Wang, M. and N. Yang (2018). “Three-dimensional computational model simulating the fracture healing process with both biphasic poroelastic finite element analysis and fuzzy logic control”. In: *Scientific Reports* 8.1, p. 6744. ISSN: 2045-2322. URL: <https://doi.org/10.1038/s41598-018-25229-7>.
- Wang, X., T.S.E. Eriksson, T. Ricken, et al. (2018). “On incorporating osmotic pre-stretch/prestress in image-driven finite element simulations of cartilage”. In: *Journal of the Mechanical Behavior of Biomedical Materials* 86, pp. 409–422. ISSN: 1751-6161. URL: <https://doi.org/10.1016/j.jmbbm.2018.06.014>.
- Wang, Y., W. Wang, Y. Liang, et al. (2021). “Mixup for Node and Graph Classification”. In: *Proceedings of the Web Conference 2021*. WWW '21. Ljubljana, Slovenia: Association for Computing Machinery, pp. 3663–3674. ISBN: 9781450383127. URL: <https://doi.org/10.1145/3442381.3449796>.
- Wang, Y., Y. Zhao, et al. (2022). “Imbalanced graph classification via graph-of-graph neural networks”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. CIKM '22. Atlanta, GA, USA: Association for Computing Machinery, pp. 2067–2076. ISBN: 9781450392365. URL: <https://doi.org/10.1145/3511808.3557356>.
- Weisbecker, H., D.M. Pierce, and G.A. Holzapfel (2014). “A generalized prestressing algorithm for finite element simulations of preloaded geometries with application to the

- aorta". In: *International Journal for Numerical Methods in Biomedical Engineering* 30.9, pp. 857–872. URL: <https://doi.org/10.1002/cnm.2632>.
- Wilson, W. et al. (2004). "Stresses in the local collagen network of articular cartilage: A poroviscoelastic fibril-reinforced finite element study". In: *Journal of Biomechanics* 37.3, pp. 357–366. ISSN: 0021-9290. URL: [https://doi.org/10.1016/S0021-9290\(03\)00267-7](https://doi.org/10.1016/S0021-9290(03)00267-7).
- Wilson, W., J.M. Huyghe, and C.C. van Donkelaar (2007). "Depth-dependent compressive equilibrium properties of articular cartilage explained by its composition". In: *Biomechanics and Modeling in Mechanobiology* 6.1, pp. 43–53. ISSN: 1617-7940. URL: <https://doi.org/10.1007/s10237-006-0044-z>.
- Wilson, W. et al. (2005). "A fibril-reinforced poroviscoelastic swelling model for articular cartilage". In: *Journal of Biomechanics* 38.6, pp. 1195–1204. ISSN: 0021-9290. URL: <https://doi.org/10.1016/j.jbiomech.2004.07.003>.
- Wu, Z., S. Pan, F. Chen, et al. (2021). "A comprehensive survey on graph neural networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1, pp. 4–24. URL: <https://doi.org/10.1109/tnnls.2020.2978386>.
- Xu, J., X. Sun, Z. Zhang, et al. (2019). "Understanding and improving layer normalization". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, et al. Vol. 32. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2019/file/2f4fe03d77724a7217006e5d16728874-Paper.pdf>.
- Xue, Z., Y. Yang, and R. Marculescu (2023). "SUGAR: Efficient subgraph-level training via resource-aware graph partitioning". In: *IEEE Transactions on Computers* 72.11, pp. 3167–3177. URL: <https://doi.org/10.1109/TC.2023.3288755>.
- Yang, Y. and P. Perdikaris (2019). "Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems". In: *Computational Mechanics* 64.2, pp. 417–434. ISSN: 1432-0924. URL: <https://doi.org/10.1007/s00466-019-01718-y>.
- Yeo, I.-K. and R.A. Johnson (2000). "A new family of power transformations to improve normality or symmetry". In: *Biometrika* 87.4, pp. 954–959. ISSN: 0006-3444. URL: <https://doi.org/10.1093/biomet/87.4.954>.
- You, Y., T. Chen, Y. Sui, et al. (2020). "Graph contrastive learning with augmentations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, et al. Vol. 33. Curran Associates, Inc., pp. 5812–5823. URL: <https://doi.org/10.48550/arXiv.2010.13902>.
- Zahn, A. and D. Balzani (2016). "Modeling of anisotropic growth and residual stresses in arterial walls". In: 7, pp. 85–90. URL: <http://dx.doi.org/10.14311/APP.2017.7.0085>.

- Zeng, H., H. Zhou, A. Srivastava, et al. (2020). “GraphSAINT: Graph sampling based inductive learning method”. In: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=BJe8pkHFwS>.
- Zhang, Hongyi et al. (2018). “mixup: beyond empirical risk minimization”. In: *International Conference on Learning Representations*. URL: <https://doi.org/10.48550/arXiv.1710.09412>.
- Zhang, L. et al. (2021). “Multi-fidelity surrogate model-assisted fatigue analysis of welded joints”. In: *Structural and Multidisciplinary Optimization* 63.6, pp. 2771–2787. ISSN: 1615-1488. URL: <https://doi.org/10.1007/s00158-020-02840-9>.
- Zhang, L., Y. Wu, P. Jiang, et al. (2022). “A multi-fidelity surrogate modeling approach for incorporating multiple non-hierarchical low-fidelity data”. In: *Advanced Engineering Informatics* 51, p. 101430. ISSN: 1474-0346. URL: <https://doi.org/10.1016/j.aei.2021.101430>.
- Zhang, X., Y. Xu, W. He, et al. (2024). “A comprehensive review of the oversmoothing in graph neural networks”. In: *Computer Supported Cooperative Work and Social Computing*. Singapore: Springer Nature Singapore, pp. 451–465. ISBN: 978-981-99-9637-7. URL: [https://doi.org/10.1007/978-981-99-9637-7\\_33](https://doi.org/10.1007/978-981-99-9637-7_33).
- Zhao, X., Z. Gong, J. Zhang, et al. (2021). “A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout”. In: *Structural and Multidisciplinary Optimization* 64.4, pp. 2287–2306. ISSN: 1615-1488. URL: <https://doi.org/10.1007/s00158-021-02983-3>.
- Zhou, J., G. Cui, S. Hu, et al. (2020). “Graph neural networks: A review of methods and applications”. In: *AI Open* 1, pp. 57–81. ISSN: 2666-6510. URL: <https://doi.org/10.1016/j.aiopen.2021.01.001>.
- Zhou, Q., P. Jiang, X. Shao, et al. (2017). “A variable fidelity information fusion method based on radial basis function”. In: *Advanced Engineering Informatics* 32, pp. 26–39. ISSN: 1474-0346. URL: <https://doi.org/10.1016/j.aei.2016.12.005>.
- Zhuang, F., Z. Qi, K. Duan, et al. (2021). “A comprehensive survey on transfer learning”. In: *Proceedings of the IEEE* 109.1, pp. 43–76. URL: <https://doi.org/10.1109/JPROC.2020.3004555>.

# Appendix A

## Runtime Details

Table A.1: Runtime details on a regular CPU for generating small-scale numerical samples used in Chapter 4.

2D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
1	0	21	2	21
2	0	21	2	21
3	0	21	2	21
4	0	21	2	21
5	0	21	2	21
6	0	21	2	21
7	0	21	2	20
8	0	21	2	19
9	0	21	2	22
10	0	21	2	21
11	0	21	2	21
12	0	21	2	21
13	0	21	2	21
14	0	21	2	21
15	0	21	2	21
16	0	21	2	21
17	0	21	2	21
18	0	21	2	22
19	0	21	2	21
20	0	21	2	22
21	0	21	2	21
22	0	21	2	22

... continued

2D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
23	0	21	2	22
24	0	21	2	22
25	0	21	2	22
26	0	21	2	21
27	0	21	2	21
28	0	21	2	22
29	0	21	2	22
30	0	21	2	22
31	0	24	2	11
32	0	18	2	10
33	0	18	2	10
34	0	18	2	10
35	0	18	2	10
36	0	19	2	11
37	0	18	2	11
38	0	18	1	51
39	0	18	2	10
40	0	18	2	11

Table A.2: Runtime details on a regular CPU for generating large-scale numerical samples used in Chapter 4.

3D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
1	1	22	10	22
2	1	5	7	31
3	1	1	7	37
4	1	51	14	23
5	1	9	11	53
6	1	7	11	40
7	1	19	9	54
8	1	11	12	2
9	1	31	9	2
10	1	23	12	19

...continued

3D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
11	1	3	7	11
12	1	15	11	27
13	1	23	8	37
14	1	53	14	22
15	1	13	7	15
16	1	25	14	50
17	1	5	4	2
18	1	17	11	40
19	1	7	7	29
20	1	5	7	18
21	0	59	7	25
22	2	3	20	48
23	1	15	12	36
24	1	29	9	57
25	1	11	7	20
26	1	21	12	14
27	1	17	7	13
28	1	5	8	7
29	1	23	13	50
30	1	11	12	58
31	1	0	7	55
32	2	13	17	15
33	1	7	7	53
34	1	25	15	43
35	1	5	7	42
36	1	9	7	25
37	1	9	10	30
38	1	3	4	18
39	1	19	10	5
40	1	23	14	18
41	1	11	10	58
42	1	17	7	55
43	1	13	11	52
44	1	29	15	39
45	1	13	13	14

...continued

3D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
46	1	7	4	36
47	1	17	19	32
48	1	19	10	28
49	1	3	7	35
50	1	7	4	14
51	1	21	14	1
52	1	21	12	36
53	1	43	13	10
54	1	23	15	5
55	1	4	9	5
56	1	23	13	38
57	2	3	21	21
58	1	21	9	39
59	1	17	10	17
60	1	7	7	25
61	1	7	8	29
62	1	3	8	48
63	1	53	14	19
64	1	19	9	4
65	1	51	13	52
66	1	43	15	7
67	1	15	13	25
68	1	9	11	31
69	1	7	7	23
70	1	33	13	3
71	1	17	18	24
72	1	43	13	36
73	1	21	9	36
74	1	15	9	18
75	1	3	7	40
76	1	23	12	21
77	1	11	10	10
78	1	19	9	50
79	2	5	13	1
80	1	17	9	8

... continued

3D Sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
81	1	21	12	17
82	1	39	14	36
83	1	7	11	31
84	1	37	13	3
85	1	7	7	8
86	1	23	9	42
87	2	17	15	32
88	1	17	8	42
89	1	13	9	58
90	1	19	8	34
91	1	5	3	54
92	1	5	8	28
93	1	7	7	7
94	1	9	10	37
95	1	11	9	54
96	1	5	7	7
97	1	1	8	8
98	2	11	15	8
99	1	1	8	2
100	1	2	4	6

Table A.3: Runtime details on a regular CPU for generating small-scale numerical samples used in Chapter 5.

Small-scale sample	Low-fidelity		High-fidelity model	
	Min	Sec	Min	Sec
1	1	1	21	29
2	1	16	18	54
3	0	56	21	53
4	0	48	19	54
5	3	56	19	51
6	1	1	20	0
7	0	56	23	4
8	1	36	18	7

... continued

Small-scale sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
9	1	31	16	44
10	0	51	16	7
11	0	46	19	39
12	0	52	21	13
13	1	21	19	38
14	1	6	18	16
15	2	46	22	21
16	1	21	19	19
17	1	31	19	53
18	0	56	21	22
19	0	51	20	13
20	0	46	23	6
21	0	52	22	27
22	1	6	20	43
23	1	32	18	10
24	1	21	19	0
25	1	6	22	52
26	1	42	24	10
27	4	12	19	10
28	1	22	20	45
29	0	46	21	20
30	1	26	24	3
31	2	1	21	32
32	1	12	21	45
33	1	41	21	41
34	1	6	27	0
35	0	56	23	56
36	0	46	21	34
37	1	1	20	55
38	1	2	22	32
39	1	8	17	42
40	0	56	24	11
41	0	52	22	54
42	1	56	22	14
43	0	58	23	9

... continued

Small-scale sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
44	0	46	21	35
45	1	11	20	20
46	0	56	22	37
47	1	11	21	6
48	1	21	21	27
49	1	21	23	45
50	1	6	24	38

Table A.4: Runtime details on a regular CPU for generating large-scale numerical samples used in Chapter 5.

Large-scale sample	Low-fidelity model		High-fidelity model	
	Min	Sec	Min	Sec
1	3	0	120	55
2	3	2	110	59
3	3	28	114	38
4	2	54	114	4
5	3	14	120	23
6	2	42	128	48

# Appendix B

## Supplemental Results

Table B.1: In-distribution mean-squared error of the small-scale model used in Chapter 5 with message passings: 2. Abbreviations: AE = autoencoder; CR = compression ratio.

Model type / with AE	CR: 4		CR 2		CR: 1		CR: 0.5	
	No	Yes	No	Yes	No	Yes	No	Yes
<b>Dynamic subgraphing</b>	0.08	0.13	0.11	0.08	0.07	0.07	0.07	0.08
<b>Static subgraphing</b>	0.15	0.17	0.09	0.25	0.10	0.15	0.10	0.99
<b>Dynamic weighting</b>	0.09	0.27	0.07	0.19	0.08	0.40	0.07	0.53
<b>Static weighting</b>	0.08	0.21	0.06	0.31	0.07	0.41	0.07	0.34
<b>Maximal loss</b>	0.08	0.17	0.07	1.10	0.08	0.76	0.08	1.44
<b><math>l_2</math> loss</b>	0.07	0.13	0.08	0.20	0.08	0.18	0.05	0.34

Table B.2: In-distribution mean-squared error of the small-scale model used in Chapter 5 with message passings: 5. Abbreviations: AE = autoencoder; CR = compression ratio.

Model type / with AE	CR: 4		CR 2		CR: 1		CR: 0.5	
	No	Yes	No	Yes	No	Yes	No	Yes
<b>Dynamic subgraphing</b>	0.06	0.10	0.08	0.09	0.17	0.06	0.10	0.07
<b>Static subgraphing</b>	0.10	0.41	0.10	0.30	0.09	0.28	0.10	1.11
<b>Dynamic weighting</b>	0.09	0.26	0.07	0.30	0.07	0.22	0.08	0.23
<b>Static weighting</b>	0.08	0.20	0.08	0.32	0.08	0.16	0.07	0.19
<b>Maximal loss</b>	0.06	1.80	0.06	0.33	0.07	7.48	0.07	1.10
<b><math>l_2</math> loss</b>	0.06	0.36	0.09	0.16	0.08	26.07	0.07	0.19

Table B.3: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 4 and number of message passings 2. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	0.13	0.10	0.45	0.16
<b>Static subgraphing</b>	0.47	0.16	1.26	0.94
<b>Dynamic weighting</b>	0.09	0.19	0.43	14.83
<b>Static weighting</b>	0.12	0.67	0.55	0.99
<b>Maximal loss</b>	0.09	0.16	0.51	19.73
<b><math>l_2</math> loss</b>	0.13	0.11	1.08	1.52

Table B.4: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 4 and number of message passings 5. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	1.28	0.15	7.42	2.52
<b>Static subgraphing</b>	0.17	0.59	18.65	2.68
<b>Dynamic weighting</b>	0.21	0.30	0.22	0.96
<b>Static weighting</b>	0.11	0.24	0.16	1.94
<b>Maximal loss</b>	0.11	2.35	0.47	0.78
<b><math>l_2</math> loss</b>	0.11	0.43	12.06	1.15

Table B.5: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 2 and number of message passings 2. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	0.84	0.09	0.62	1.22
<b>Static subgraphing</b>	0.25	0.53	0.71	4.61
<b>Dynamic weighting</b>	0.15	0.37	0.40	0.80
<b>Static weighting</b>	0.13	0.50	2.05	0.72
<b>Maximal loss</b>	0.09	0.79	1.87	0.52
<b><math>l_2</math> loss</b>	0.11	0.24	0.55	2.32

Table B.6: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 2 and number of message passings 5. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	0.09	0.14	66.11	56.07
<b>Static subgraphing</b>	2.83	2.62	163.23	100.30
<b>Dynamic weighting</b>	0.10	0.34	1.69	0.89
<b>Static weighting</b>	0.25	0.28	0.99	0.51
<b>Maximal loss</b>	0.10	0.20	3018.96	1279.40
<b><math>l_2</math> loss</b>	0.39	4.29	26.25	1299.88

Table B.7: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 1 and number of message passings 2. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	0.14	0.33	0.97	0.65
<b>Static subgraphing</b>	0.33	0.08	0.44	1.80
<b>Dynamic weighting</b>	0.13	0.46	0.88	0.86
<b>Static weighting</b>	0.12	0.47	0.59	0.99
<b>Maximal loss</b>	0.13	0.90	6.06	68.72
<b><math>l_2</math> loss</b>	0.09	0.19	0.44	0.53

Table B.8: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 1 and number of message passings 5. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
<b>Dynamic subgraphing</b>	0.78	0.72	1159.24	25001.22
<b>Static subgraphing</b>	37.67	15.85	5983.59	3685.13
<b>Dynamic weighting</b>	0.17	0.20	0.41	0.84
<b>Static weighting</b>	0.11	0.17	0.76	0.61
<b>Maximal loss</b>	0.12	0.82	12.27	554.70
<b><math>l_2</math> loss</b>	0.89	15.37	2.90	792.63

Table B.9: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 0.5 and number of message passings 2. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
Dynamic subgraphing	0.58	0.77	29.05	2.83
Static subgraphing	0.18	0.85	6.17	3.51
Dynamic weighting	0.10	0.68	0.68	0.82
Static weighting	0.11	0.36	0.48	0.78
Maximal loss	0.07	1.14	35.30	5.81
$l_2$ loss	0.16	7.86	1.05	1.05

Table B.10: Out-of-distribution mean-squared error of the full model used in Chapter 5 with compression ratio 0.5 and number of message passings 5. Abbreviation: AE = autoencoder.

Model type	With augmentation		Without augmentation	
	With AE	Without AE	With AE	Without AE
Dynamic subgraphing	0.58	0.25	276541.50	46.19
Static subgraphing	0.75	6.94	29.73	118.65
Dynamic weighting	0.17	0.27	3.32	0.37
Static weighting	0.13	0.18	4.06	1.72
Maximal loss	0.08	1.92	60.22	94.65
$l_2$ loss	0.25	0.32	250.52	76.40

