Emulation of Cardiac Mechanics Using E(3) Equivariant Graph Neural Networks

Adrien Pinard[1], Zhinuo Jenny Wang[1], Ambre Bertrand[1], Blanca Rodriguez[1], Julia Camps[1,2]

[1]University of Oxford, Oxford, United Kingdom
[2]Universitat Pompeu Fabra, Barcelona, Spain

Abstract

Finite-element-based simulations of cardiac electromechanics, though accurate, remain prohibitively computationally expensive, often requiring hours per beat using high-performance computing resources. We present a deep learning-based emulator leveraging E(3) equivariant graph neural networks (GNNs) to approximate cardiac passive mechanics, enabling fast and generalizable predictions of myocardial deformations. Our architecture encodes both geometric and physiological features in an E(3)-equivariant form and introduces a multi-resolution graph augmentation strategy to model long-range dependencies. The proposed method achieves substantial acceleration while maintaining good accuracy.

1. Introduction

Simulating the human heart's electromechanics using finite-element methods requires several hours per beat on a supercomputer. A graph neural network (GNN) model trained on simulated data can approximate the cardiac passive mechanics and speed up these calculations [1]. However, the machine learning model must be E(3) equivariant to generalise to different geometries. Here, we propose an E(3) equivariant extension of [1].

2. Methods

Our GNN model takes as input the resting biventricular geometry as a graph G and its fibre orientations, which are represented using ventricular coordinates [2] combined with the strategies in [3] to achieve E(3) equivariance. The model then predicts the node-wise displacements produced by the passive mechanics of the diastolic filling phase.

2.1. E(3) Invariant Inputs

The cardiac passive mechanics simulation problem is equivariant to rotations, translations and reflections. To enforce these equivariances, we use an E(3) invariant representation of the input data. The inputs that were not equivariant in their original representation were the node coordinates and the fibre orientations. The GNN automatically provides an invariant embedding for the node coordinates, thus leaving only the fibre orientations.

To generate an invariant representation of the feature vectors (i.e., fibre orientations), [4] proposes to project each vector on a local basis defined for each edge by $B_{i,j} = (\overrightarrow{a_{i,j}}, \overrightarrow{b_{i,j}}, \overrightarrow{c_{i,j}})$. Such that:

$$\overrightarrow{a_{i,j}} = \overrightarrow{x_i} - \overrightarrow{x_j}$$
$$\overrightarrow{b_{i,j}} = \frac{\overrightarrow{x_i} \times \overrightarrow{x_j}}{||\overrightarrow{x_i} \times \overrightarrow{x_j}||}$$
$$\overrightarrow{c_{i,j}} = \overrightarrow{a_{i,j}} \times \overrightarrow{b_{i,j}}$$

This gives us a way to create local basis vectors that are equivariant regarding rotations but not translations. We extend this idea by using ventricular coordinates [2] to compute invariant local bases. In particular, we use the apex-to-base and transmural coordinates, and their cross-product. Then, we project the fibre orientations on these local basis vectors to obtain an invariant embedding.

We also include the ventricular coordinates as features to help the GNN interpret the projected fibre vectors. This leads to the embedding of the following 28 features:

- Four node-wise ventricular coordinates.
- Three invariant vectors: fibre, sheet and sheet-normal directions.
- Four node-wise Boolean masks to identify nodes on the boundaries of the geometry: RV-endocardium, LV-endocardium, epicardial, and valve.
- Eleven graph-wise parameters from the passive mechanics formulation.

2.2. GNN Architecture

We propose a message-passing GNN architecture composed of a "Features Embedding" (Linear) layer followed by k "Equivariant Layers".

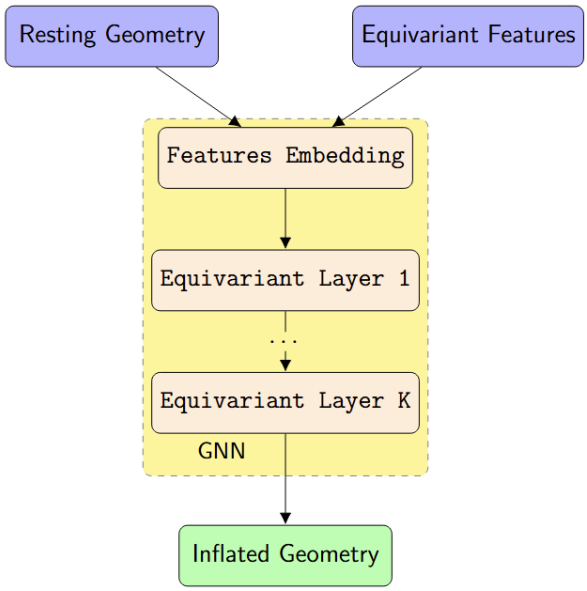


Figure 1. Model architecture schematic.

2.3. E(3) Equivariant Layers

Our approach extends the GNN strategy from [1] to make the machine learning model E(3) equivariant. To this end, we implement the layering in Figure 2 [3], combined with a novel E(3) invariant data representation.

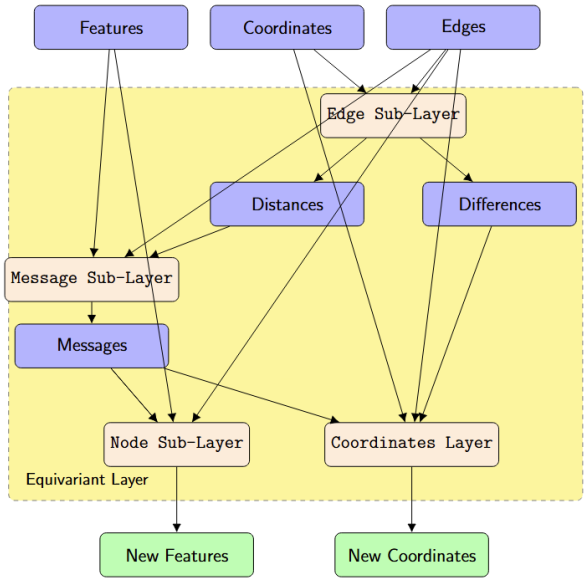


Figure 2. E(3) Equivariant layer architecture.

Edge-wise directed messages m_{ij}^l are computed from invariant features and edge attributes:

$$m_{ij}^l = \phi_m^l\left(h_i^l, h_j^l, ||\overrightarrow{x_i^l} - \overrightarrow{x_j^l}||\right)$$

Here m_{ij}^l is the directed message from node i to node j at the l-th Equivariant layer in the GNN, h_i^l are the features of node i, $\overrightarrow{x_i^l}$ is the coordinates of node i at the l-th Equivariant layer in the GNN, and ϕ_m^l, ϕ_c^l and ϕ_n^l are Linear layers corresponding respectively to the message, coordinate and node layers.

The coordinates of each node i update as a weighted sum over the messages of their adjacent nodes $j \in \mathcal{A}(i)$:

$$\overrightarrow{x_i^{l+1}} = \overrightarrow{x_i^l} + \frac{1}{\#\mathcal{A}(i)} \sum_{j \in \mathcal{A}(i)} \left(\overrightarrow{x_i^l} - \overrightarrow{x_j^l}\right) \phi_c^l(m_{ij}^l)$$

Node embeddings h_i^l are updated using the aggregated messages of each node i:

$$\mathrm{h_i^{l+1}} = \phi_\mathrm{n}^\mathrm{l}\left(\mathrm{h_i^l}, m_i^l = \frac{1}{\#\mathcal{A}(i)} \sum_{j \in \mathcal{A}(i)} m_{ij}^l\right)$$

3. Experiments and Results

We trained the GNNs on finite element method simulated deformation fields using fixed and varying pressure conditions. Loss was computed via mean squared error (MSE) between predicted and simulated node-wise displacements.

Training used various combinations of layers, hidden dimensions, graph augmentations, and dropout regularisation. Evaluation includes both absolute and relative errors normalised by maximal node displacement. We repeated each experiment ten times with different initialisations and preserved the best three for the results.

3.1. Expressivity of the GNN

This experiment aims to choose a configuration capable of representing the output displacements as a function of the inputs. To do this, we simultaneously vary both the hidden dimensions and depth of the model and evaluate the training error over a single input-output case.

The GNNs with a hidden dimension of 32 neurons and a depth of 8 layers (Figure 3) accurately learned deformations from a single simulation, achieving final MSE values of 0.013 cm. This loss was significantly lower compared to the loss obtained between the initial geometry and target geometry, 0.036 cm.

Local deformations were well-captured overall (Figure 4). However, their accuracy degraded near the basal plane due to changes in boundary conditions and material properties in those regions. From here on, we proceed with

this GNN architecture composed of 8 hidden layers of 32 neurons.

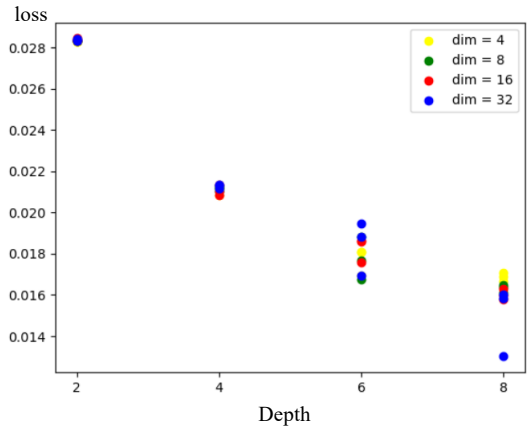


Figure 3. Model loss as a function of hidden dimensions of the model for a single geometry. The three model losses (y-axis) obtained for the overfitting (expressivity analysis) cases for each set of depth (x-axis) and hidden dimension (legend) of the hidden layers.

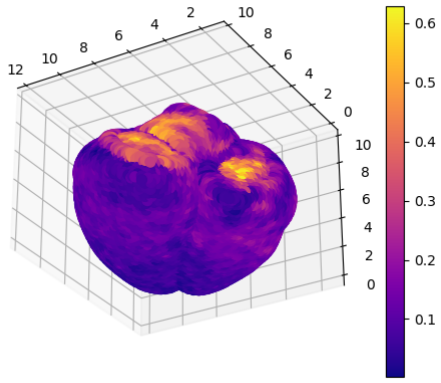


Figure 4. Relative displacement error = $|d - d'|/d$, where d and d' are the true and predicted displacement values for the base architecture with hidden dimension 32 and depth 8 on a single geometry for the overfitting case. The colorbar shows the error normalised by the maximal displacement. Coordinates are displayed in centimetres.

3.2. Effect of Graph Augmentation

Graph augmentation, similar to [1], with two coarsening levels (e.g., [1000, 20] nodes) improved training stability, particularly in shallow models (Figure 5). These results supported the hypothesis that augmenting the message-passing structure enables more effective long-range communication. From here on, we proceed with the GNN with graph augmentation [1000, 20].

loss

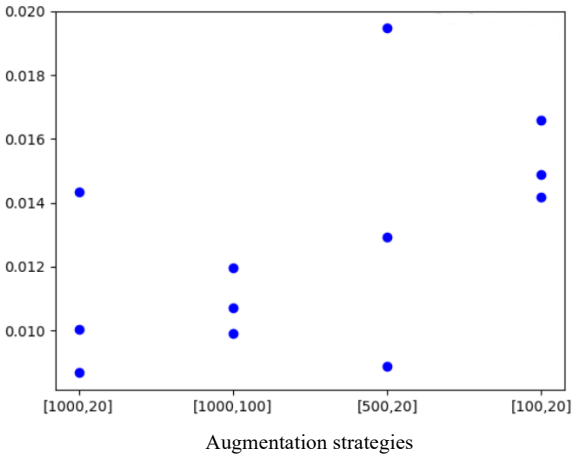


Figure 5. Model loss as a function of the augmentation parameters for a single geometry. The three model losses (y-axis) obtained for the overfitting (expressivity analysis) case for each set of augmentation strategies (x-axis).

3.3. Learning Changes in Pressure

With increments in the internal pressure ranging from 0 to 3 kPa, the model generalised reasonably well across simulations (Figure 6). Incorporating pressure features as node-wise inputs helped encode the global state.

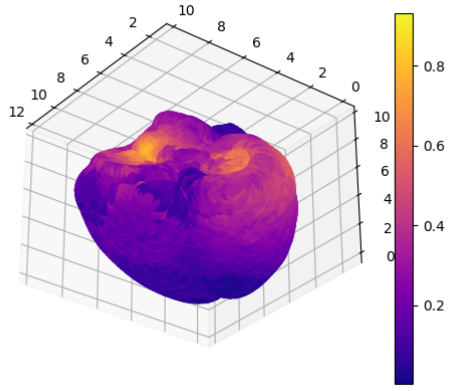


Figure 6. Relative training error (normalised by the maximal displacement) from experiment training with varying internal pressure. Coordinates are in centimetres.

3.4. Regularisation Strategies

Adding dropout improved training stability (Figure 7). We suggest at least a 20% dropout for this architecture.

4. Discussion

We tested the training capacity and prediction cost of various GNNs on one geometry to emulate cardiac passive mechanics. The GNNs predicted the passive deformations in ~5 seconds on a desktop machine, resulting in a speed-

up of 3 orders of magnitude compared to the finite-element methods solver used for generating the training data.

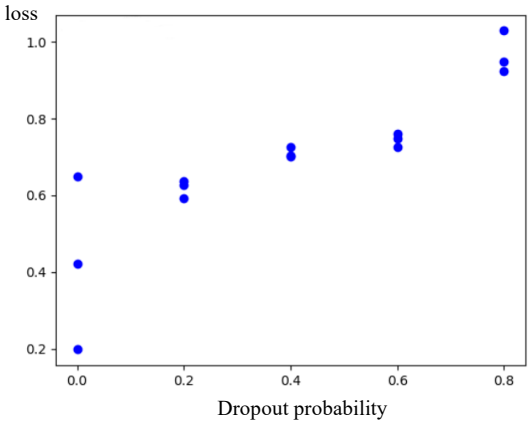


Figure 7. Model loss as a function of the dropout ratio on a set of samples with various internal pressures. The three model losses (y-axis) obtained for each value of dropout ratio (x-axis).

Our results have also illustrated the advantages and limitations of GNNs for accelerating passive cardiac mechanics simulations with homogeneous tissue properties.

We have also shown that implementing strategies as graph augmentation and regularisation can improve the stability of the training of GNNs.

We observed high correlations on the prediction errors in regions with discontinuities in the description of the problem, such as different material and contractile properties (i.e., the valvular plugs), as well as changing boundary conditions (i.e., the base-to-valves region). More sophisticated strategies for graph augmentation that consider regional discontinuities could potentially improve the model's predictions. These errors may relate to limitations of the reference finite element method formulation in replicating physiological behaviour in these regions. New strategies to represent these discontinuities comprehensively could improve the model's predictions.

While our model uses a single geometry with varying pressures, generalising to different cardiac anatomies, material properties, and boundary conditions remains future work. Additional improvements could involve using a different representation for the boundary conditions and incorporating physics-informed losses.

5. Conclusions

We presented a symmetry-aware deep learning framework for emulating cardiac passive mechanics. By combining GNNs with cardiac coordinate-based E(3) invariant embeddings and hierarchical graph augmentations, our model replicates FE simulations with high accuracy and reduced compute time. This offers a promising path toward scalable cardiac digital twins.

Acknowledgments

The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434) to JC (LCF/BQ/PI25/12100029), a Wellcome Trust Fellowship in Basic Biomedical Sciences to BR (214290/Z/18/Z), an EPSRC Centre for Doctoral Training in Health Data Science scholarship to AB (EP/S02428X/1), the PRACE ICEI project icp019 to ZJW, the CompBioMed X (EP/X019446/1) and the CompBioMed 2 (EC Horizon 2020 R&I programme, grant agreements 675451 and 823712, respectively).

We thank Dr Mariano Vázquez and the other members of ELEM for providing access to the Alya software.

This study is published under a Creative Commons Attribution (CC BY) public copyright licence.

References

[1] D. Dalton, H. Gao, and D. Husmeier, 'Emulation of cardiac mechanics using graph neural networks', *Comput. Methods Appl. Mech. Eng.*, vol. 401, p. 115645, Nov. 2022, doi: 10.1016/j.cma.2022.115645.

[2] J. Bayer *et al.*, 'Universal ventricular coordinates: A generic framework for describing position within the heart and transferring data', *Med. Image Anal.*, vol. 45, pp. 83–93, Apr. 2018, doi: 10.1016/j.media.2018.01.005.

[3] V. G. Satorras, E. Hoogeboom, and M. Welling, 'E(n) equivariant graph neural networks', Feb. 16, 2022, *arXiv*: arXiv:2102.09844. doi: 10.48550/arXiv.2102.09844.

[4] W. Du *et al.*, 'SE(3) equivariant graph neural networks with complete local frames', Jul. 06, 2022, *arXiv*: arXiv:2110.14811. doi: 10.48550/arXiv.2110.14811.

Address for correspondence:

Julia Camps.
Tanger building, Tanger, 122-140, Barcelona, Spain.
julcamps@gmail.com.