ORIGINAL PAPER

Thermodynamics-informed graph neural networks for real-time simulation of digital human twins

Lucas Tesán[1] · David González[1] · Pedro Martins[1,2] · Elías Cueto[1]

Abstract

The growing importance of real-time simulation in the medical field has exposed the limitations and bottlenecks inherent in the digital representation of complex biological systems. This paper presents a novel methodology aimed at advancing current lines of research in soft tissue simulation. The proposed approach introduces a hybrid model that integrates the geometric bias of graph neural networks with the physical bias derived from the imposition of a metriplectic structure as soft and hard constrains in the architecture, being able to simulate hepatic tissue with dissipative properties. This approach provides an efficient solution capable of generating predictions at high feedback rate while maintaining a remarkable generalization ability for previously unseen anatomies. This makes these features particularly relevant in the context of precision medicine and haptic rendering. Furthermore, this work synthesizes two prominent concepts in recent years: the role of message passing as a geometric mechanism fundamental to graph neural networks, and the potential of thermodynamics-informed networks to enhance extrapolation capabilities beyond training scenarios. We develop a multi-graph interaction between the computational model of the liver and a surgical tool. A displacement imposed at the contact region initiates a controlled flow of information that propagates throughout the graph model, aiming to achieve a steady and more efficient exchange of information across the entire network. The physics bias is obtained by imposing a metriplectic structure, enforced via strong and soft constraints. This ensures that the network satisfies thermodynamic principles during inference, even for a previously unseen system. Based on the adopted methodologies, we propose a model that predicts human liver responses to traction and compression loads in as little as 7.3 milliseconds for optimized configurations and as fast as 1.65 milliseconds in the most efficient cases, all in the forward pass. The model achieves relative position errors below 0.15%, with stress tensor and velocity estimations maintaining relative errors under 7%. This demonstrates the robustness of the approach developed, which is capable of handling diverse load states and anatomies effectively. This work highlights the feasibility of integrating real-time simulation with patient-specific geometries through deep learning, paving the way for more robust digital human twins in medical applications.

Keywords Deep learning · Graph neural networks · Thermodynamics · GENERIC · Biomechanics · Soft tissue · Digital twins.

1 Introduction

In recent decades, real-time simulation has emerged as a fundamental tool for modeling complex systems in diverse fields such as engineering, biology, physics, and social sciences.

✉ Pedro Martins
pmartins@unizar.es

[1] ESI Group-UZ Chair of the National Strategy on Artificial Intelligence, Aragon Institute of Engineering Research (I3A), Universidad de Zaragoza, Zaragoza, Spain

[2] Aragonese Foundation for Research and Development (ARAID), Aragón, Spain

These systems, which range from structural mechanics [1] to climate models [2] and even biological systems themselves [3–6], require significant computational power to ensure accuracy and efficiency in real-time decision-making.

In the context of this work, the primary focus is on the computational mechanics involved in modeling the response of a soft tissue, whose viscous-hyperelastic characteristics have long posed a classical challenge for established numerical methods [7–9]. The issue lies in the mesh dependency of these methodologies [10, 11], where both error and computational time are nonlinearly dependent on the element size,

order, and shape factor, significantly constraining the model's quality in environments that require high feedback rates.

Thus, the numerical methodology is based on iterative convergence towards the real solution, defined by the physical laws governing the model, and it is only limited by the spatio-temporal discretizations. In contrast, data-driven approaches, usual in general machine learning, achieve convergence towards ground truth values largely based on the amount and diversity of data used for training. Under these conditions, inference is heavily influenced by the hyperspace of conditions defined in the training set, with hyperparameter tuning of the network playing a secondary, but relevant, role in determining the model performance.

As a result of these problems, unpredictability of so called *black-box* dense models, new lines of research have emerged that propose alternatives to mitigate this pronounced bias, constraining the variability of possible solutions to those that best represent the physical reality defined by the underlying problem: geometric bias, widely popularized through the use of convolutional neural networks (CNNs) [12], defines and controls the mathematical structure of the optimized solution by leveraging the geometric properties of the data [13]. In the case of CNNs, this translates into the localization of the pixels that make up an image and their positional relationships. In recurrent networks, the geometric bias manifests in the temporal dependency between the data [14], while in graph networks, a prior relationship is established before training. This prior describes the direction and extent of the information flow, although the exact effect of this connectivity is unknown [15, 16].

On the other hand, physical biases assert that the data follow a differential structure governed by the physical laws that regulate the modeled process. These biases allow us to establish weak constraints during loss optimization so that the residual of a differential equation or a degeneration term is satisfied, consequently aligning the model with the underlying physical principles and defining the so-called physics-informed neural networks (PINNs)[17].

In this way, when all this prior human knowledge translated into learning biases strengthens the predictive capacity of the network, a hybrid artificial intelligence is generated [18], forming the conceptual backbone of the current work, as shown in Fig. 1.

1.1 Objective

The main objective of this work is to develop a robust hybrid neural simulator for real-time inference of the response of a patient-specific hepatic twin. As additional goals we also consider the validation of physical inductive biases through comparative analysis with data-only models. Additionally, it seeks to optimize convergence in message-passing processes to enhance response speed, ensuring a comprehensive flow

of information across all nodes within the network's latent space. Finally, this study will analyze the model sensitivity in scenarios characterized by reduced training datasets and scarce data.

1.2 Previous work

As we have pointed out, PINNs are a robust methodology for tackling systems whose governing partial differential equations are known. However, in many cases, this information is not available, or at least not completely, especially considering the natural complexity of biological systems. In the absence of knowing a system's governing equations, the establishment of a parallel between the learning of the physical phenomenon and that of a dynamical system is of great help in formulating the problem as a non-linear regression, that will provide an expression for the time evolution of the system under study [19].

This approach enables the incorporation of specific physical properties into the learning process, ensuring compliance with natural first principles. For example, it allows learned simulators to uphold energy conservation within their predictions, an ability that has sparked considerable interest and was thoroughly investigated. Many studies, particularly on Hamiltonian [20, 21] and Lagrangian neural networks [22, 23], leverage these frameworks to ensure energy conservation during inference. However, a primary limitation of this method is that most physical processes are dissipative and irreversible at larger scales, requiring further conditions to satisfy both the first and second laws of thermodynamics.

Our work advances this field by introducing a novel approach that combines deep learning, mesh-based simulations, and physics-informed inductive biases, exploiting recent advancements in these fields [24, 25]. Drawing on principles from geometric deep learning, we develop a method that models a system's physical interactions with external loads within a multi-graph framework [15], enforcing thermodynamic consistency through specialized inductive biases. This framework facilitates the hybrid learning of the dynamics from diverse physical systems, while ensuring spatial equivariance and scalability across varying resolutions [26]. While previous applications of graph neural networks to the neural simulation of biological systems exist, to the best of our knowledge, none of them employs physics-based inductive biases [27].

Furthermore, recent works that consider the region represented by each node as an open system are taken into account [28], employing a modification of the metriplectic formalism based on the port-Hamiltonian formalism [29]. This approach facilitates the development of a robust framework for the real-time simulation of soft tissue dynamics, establishing an innovative pathway for computing the mechanical response of the system that integrates precision medicine and real-time

Fig. 1 Comparison between the main characteristics of numerical methods, hybrid AI, and black box models
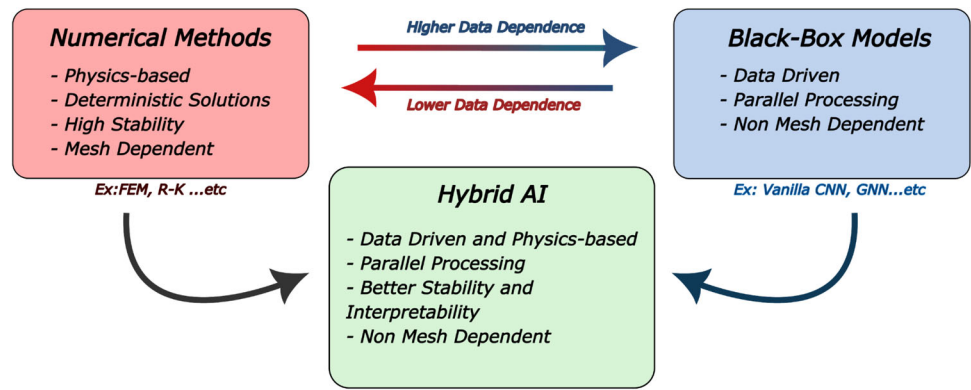
prediction. This approach is opening a pathway towards the creation of foundational biological models able to achieve an appropriate balance between the amount of data utilized and the performance of the imposed biases [30].

However, although the utility of physics-based hybrid learning for computational mechanics has been demonstrated [31, 32], a bottleneck remains concerning graph networks, whose limitations arise from the computational complexity of the model. Large-scale simulations with fine meshes require substantial message passing to convey information across the entire domain [33], or a very fine time discretization, both of which significantly increase computational cost.

2 Methodology

2.1 Problem statement

Let $z \in \mathcal{M} \subseteq \mathbb{R}^n$ represent the independent state variables that fully characterize the dynamical system at a certain level of description. The set $\mathcal{M}$, known as the state space, contains all admissible states and is assumed to have the structure of a differentiable manifold in $\mathbb{R}^n$. The physical phenomenon under consideration can be modeled by a system of ordinary differential equations that govern the time evolution of each state variable z [24],

$$\dot{z} = \frac{\mathrm{d}z}{\mathrm{d}t} = F(z, t), \ t \in \mathcal{I} = (0, T], \ z(0) = z_0, \tag{1}$$

where t refers to the time coordinate in the time interval $\mathcal{I}$ and $F(z, t)$ refers to an arbitrary nonlinear function whose precise form is to be sought from data.

The main objective of this work is, given a time discretization consisting of N time steps, to determine the function F such that, for a given temporal state z^t, its temporal evolution can be calculated and, consequently, its next temporal state z^{t+1} [19]. For this purpose, a temporal iterative scheme

based on forward Euler integration is defined,

$$z^{t+1} = z^t + \Delta t \cdot F(z^t), \tag{2}$$

wherein the function $F(z^t)$ is represented by a deep neural model $\mathcal{N}$, which depends solely on the input z^t and is parameterized by $\boldsymbol{\theta}$ (the weights and biases of the network),

$$F(z^t) = \mathcal{N}(z^t; \boldsymbol{\theta}). \tag{3}$$

2.2 Metriplectic structure: the GENERIC formalism

To ensure that the resulting $F(z^t)$-function conforms to the physical principles known in the scientific literature, it will be imposed in this work that it must have a metriplectic structure.

This is guaranteed by enforcing that F adheres to the so-called General Equation for Non-Equilibrium Reversible-Irreversible Coupling (GENERIC) formalism [34]. This framework extends the classical Hamiltonian approach to include dissipative systems, offering an appealing alternative for describing soft tissue dynamics while ensuring thermodynamic correctness by construction [35].

Equivalently to what is described in Eq. (1), the GENERIC framework models the system's temporal evolution, $\dot{z}$, governed through a set of state variables. It ensures that the resulting dynamic evolution satisfies fundamental thermodynamic principles throughout the process. The GENERIC formalism is composed by two blocks,

$$\dot{z} = \{z, E\} + [z, S], \tag{4}$$

where $\{\cdot, \cdot\}$ denotes the classic Poisson bracket and $[\cdot, \cdot]$ represents the dissipative bracket. To facilitate practical use, these brackets are often reformulated using two linear operators:

$$\boldsymbol{L} : T^*\mathcal{M} \to T\mathcal{M}, \quad \boldsymbol{M} : T^*\mathcal{M} \to T\mathcal{M}, \tag{5}$$

where $T^*\mathcal{M}$ and $T\mathcal{M}$ denote the cotangent and tangent bundles of the state space $\mathcal{M}$, respectively. The operator

$L = L(z)$ corresponds to the Poisson bracket and must be skew-symmetric. Similarly, the friction matrix $M = M(z)$ captures the irreversible behavior of the system and is symmetric and positive semi-definite, ensuring a non-negative dissipation rate [24].

Given these components and considering a system discretized into n_v particles, the model must adhere to the laws of thermodynamics. To enforce this, we assume a metriplectic evolution [36, 37], whose precise form is to be found from data:

$$\dot{z} = L(z)\frac{\partial E}{\partial z} + M(z)\frac{\partial S}{\partial z}, \tag{6}$$

where $z \in \mathbb{R}^{n_v \times n_{\mathrm{dof}}}$ and n_{dof} represents the number of degrees of freedom of each particle.

The degeneracy conditions,

$$L(z)\frac{\partial S}{\partial z} = \mathbf{0}, \tag{7}$$

and

$$M(z)\frac{\partial E}{\partial z} = \mathbf{0}, \tag{8}$$

along with the non-negativity of the irreversible bracket, ensure the adherence to the first (energy conservation) and second (entropy inequality) laws of thermodynamics. These can be expressed mathematically as

$$\frac{dE}{dt} = \{E, E\} = 0, \tag{9}$$

and

$$\frac{dS}{dt} = [S, S] \geq 0. \tag{10}$$

For a detailed proof of this assertion, the interested reader is directed to, for instance, [35], among many other possible references in the field.

2.2.1 Nodal GENERIC implementation with a port-metriplectic formalism

Thermodynamics assumes that a closed system does not exchange energy with the outside. In purity, this is not the case for a system such as the liver, which is subject to metabolic input from the body. However, at the time scale considered in this work (on the order of a certain number of seconds), the hypothesis that the liver is a closed system seems reasonable. The extension of the GENERIC formalism to open systems offers, however, no major conceptual difficulties, should it be deemed necessary [38].

Considering the liver as a closed system under a graph discretization approach poses, however, significant challenges. The main practical difficulty of this formalism lies in the assembly of the global matrices L and M, whose size can be very large, depending on the number of particles in the discretisation. This breaks the local character of the equations, involving a considerable expenditure of memory and computational time in the computation and subsequent storage of these matrices.

In this paper we consider a reformulation of the GENERIC formalism in which each of the nodes forming the discretisation of the system (the liver) is considered as an open system that exchanges energy with its neighbours. Therefore, a reformulation of GENERIC will be done through the port-Hamiltonian framework for open systems [28, 29].

In this framework, each node symbolizes a portion of the bulk material, facilitating energy exchange with its neighboring nodes through designated ports, which correspond to the edges of the graph. In this context, the Poisson and dissipative brackets can be expressed as follows:

$$\{\cdot, \cdot\} = \{\cdot, \cdot\}_{\mathrm{bulk}} + \{\cdot, \cdot\}_{\mathrm{bound}}, \tag{11}$$

and,

$$[\cdot, \cdot] = [\cdot, \cdot]_{\mathrm{bulk}} + [\cdot, \cdot]_{\mathrm{bound}}. \tag{12}$$

In other words, both brackets are decomposed additively into bulk and boundary contributions. With this decomposition in mind, the GENERIC principle, Eq. (4) applies to the bulk portion of the system associated to each node, that now is rewritten as

$$\dot{z} = \{z, E\}_{\mathrm{bulk}} + [z, S]_{\mathrm{bulk}} = \{z, E\} + [z, S] - \{z, E\}_{\mathrm{bound}} - [z, S]_{\mathrm{bound}}. \tag{13}$$

In matrix form, we obtain

$$\dot{z} = L\frac{\partial E}{\partial z} + M\frac{\partial S}{\partial z} - L_{\mathrm{bound}}\frac{\partial E_{\mathrm{bound}}}{\partial z} - M_{\mathrm{bound}}\frac{\partial S_{\mathrm{bound}}}{\partial z}. \tag{14}$$

In this case, the degeneracy conditions hold at the bulk level only as,

$$L_{\mathrm{bulk}}(z)\frac{\partial S_{\mathrm{bulk}}}{\partial z} = \mathbf{0}, \tag{15}$$

and

$$M_{\mathrm{bulk}}(z)\frac{\partial E_{\mathrm{bulk}}}{\partial z} = \mathbf{0}, \tag{16}$$

where the subscript "bulk" refers to the fact that the present equation applies only to the system as a whole, but not to the contributions coming through its boundary. So if we plan

to impose a GENERIC-like structure at a particle level, we must assume that this particle receives energy input/outputs from surrounding particles, as dictated by the graph structure [28].

Assume that a given particle i with state variables $z_i \in \mathbb{R}^{n_{\mathrm{dof}}}$ is connected in the graph with $j = 1, \ldots, n_{\mathrm{neigh}}$ neighbouring particles. Its state variables will thus evolve in time as

$$
\begin{aligned}
\dot{z}_i = \ & L_i(z_i)\frac{\partial e_i}{\partial z_i} + M_i(z_i)\frac{\partial s_i}{\partial z_i} \\
& - \sum_{j=1}^{n_{\mathrm{neigh}}} \left(L_{ij}(z_j)\frac{\partial e_j}{\partial z_j} + M_{ij}(z_j)\frac{\partial s_j}{\partial z_j} \right),
\end{aligned}
\tag{17}
$$

where L_i and M_i act here as bulk magnitudes for particle i, and their respective boundary contributions from neighbouring particle j are referred to as L_{ij} and M_{ij}, respectively.

The total energy and entropy of the system are obtained by assuming

$$
E = \sum_{i=1}^{n_v} e_i,
\tag{18}
$$

being e_i the energy contribution per node, an

$$
S = \sum_{i=1}^{n_v} s_i.
\tag{19}
$$

where s_i is equivalently the entropy contribution per node. The degeneracy conditions at the bulk level play here the role of the particle level, so that

$$
L_i(z_i)\frac{\partial s_i}{\partial z_i} = \mathbf{0},
\tag{20}
$$

and

$$
M_i(z_i)\frac{\partial e_i}{\partial z_i} = \mathbf{0},
\tag{21}
$$

for all $i = 1, \ldots, n_v$. Defining as a whole the imposition of the metriplectic structure for the hybrid framework, schematically shown in the Fig. 4.

This modification of the GENERIC formalism ensures compliance with the equations of thermodynamics and, at the same time, avoids the assembly of the global matrices, leaving the problem as the determination, from the available data, of the precise shape of the GENERIC structure at the scale of each model node.

2.3 Geometric structure: graph neural networks

Apart from respecting the dictates of the laws of thermodynamics, the other major ingredient of this work is the desire to develop a technique that allows for the particularisation of predictions for anatomies not previously seen in training. This task is addressed by using the geometric biases present in graph neural networks [13].

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{u})$ be a directed graph, i.e., one in which the edges have a direction, where $\mathcal{V} = \{1, \ldots, n\}$ is a set of $|\mathcal{V}| = n_v$ vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of $|\mathcal{E}| = n_e$ edges and $\boldsymbol{u} \in \mathbb{R}^{F_g}$, a feature vector, where F_g denotes the dimension of this global feature vector, containing global parameters of the simulation (such as density, non-dimensional parameters, etc.). Each vertex and edge in the graph is associated with a node and a pairwise interaction between nodes respectively in a discretized physical system. To each vertex $i \in \mathcal{V}$ we associate a feature vector $\boldsymbol{v}_i \in \mathbb{R}^{F_v}$, where F_v represents its dimensionality. Basically, the method does not use the physical degrees of freedom of each node, as the finite element method would do, but its projection to a latent space, in this case of higher dimensionality, in which it is more convenient to characterise its evolution. By decoding, these values in the latent space will be returned back to the physical space. Similarly, for each edge $(i, j) \in \mathcal{E}$ we associate an edge feature vector $\boldsymbol{e}_{ij} \in \mathbb{R}^{F_e}$ [24]. In this framework, the collection of graphs $\{\mathcal{G}_1, \ldots, \mathcal{G}_M\}$ together builds the overall multi-graph G_k, with $k = 1, \ldots, M$.

Figure 2 illustrates a multi-graph example of a simple mesh composed of two hexahedral elements. The spatial discretization forms the central graph, shown in black, while an actuator graph, shown in red, is originated at the loaded node. Together, these graphs provide a more detailed and richer representation of the system.

The implementation of multi-graphs in this context allows for complementing the central graph $\mathcal{G}^m$, which represents the finite element mesh, by adding new subgraphs $\mathcal{G}^w$, generated from the loaded node. The purpose of these new subgraphs, connected to the entire domain Ω, is to enable the transmission of information instantly, starting from the loaded position. In a sense, they inform the whole mesh of the presence of a loaded node.

In the central graph, the edges $\boldsymbol{e}_{ij}^m$ collect the information corresponding to the deformation. This information will include both the relative distance during the input time step, denoted as $\boldsymbol{q}_{ij}(z^t)$ and the relative distance in the initial unloaded state $\boldsymbol{q}_{ij}(z^0)$. Similarly, the nodes $\boldsymbol{v}_i^m$ will store an embedding vector with velocity $\boldsymbol{v}_i(z^t)$, stress state $\boldsymbol{\sigma}(z^t)$, and flags $\boldsymbol{n}_i$ indicating whether a node is fixed, loaded, or in a neutral state by default.

Fig. 2 Graph representations of the unidirected actuator graph, $\mathcal{G}^w$ (in red) and the bidirected central/mesh graph, $\mathcal{G}^m$ (in black). In each graph, $\boldsymbol{v}_i$ represents node attributes, $\boldsymbol{e}_{ij}$ denotes the edge attributes, $\bar{\boldsymbol{u}}_i$ corresponds to external displacements or interactions. Thus, the complete representation of this example would be defined as the superposition of both graphs, forming the multi-graph system G
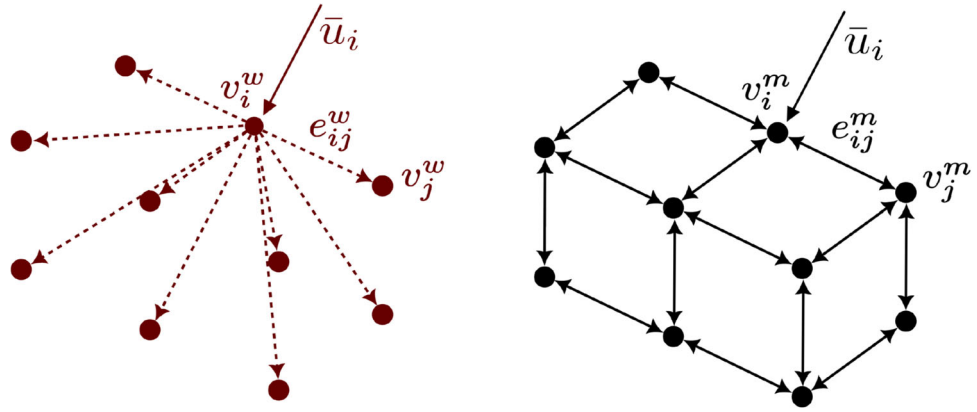
In the actuator graphs, equivalent information will be encoded in the edges $\boldsymbol{e}_{ij}^w$; however, the vertices $\boldsymbol{v}_i^w$ will contain data regarding the imposed displacement $\bar{\boldsymbol{u}}_i$.

2.3.1 Encoding

For the initial encoding structure ϵ we use four MLPs (Multi Layer Perceptron), $\epsilon = \{\epsilon_e^m, \epsilon_e^w, \epsilon_v^m, \epsilon_v^w\}$, to transform the vertex and edge initial feature vectors into the latent space, two for the central graph attributes, $\boldsymbol{x}_i^w \in \mathbb{R}^{F_l}$ and $\boldsymbol{x}_{ij}^w \in \mathbb{R}^{F_l}$ and two for the actuators graphs respectively, $\boldsymbol{x}_i^m \in \mathbb{R}^{F_l}$ and $\boldsymbol{x}_{ij}^m \in \mathbb{R}^{F_l}$.

$$\epsilon_e : \mathbb{R}^{F_e} \longrightarrow \mathbb{R}^{F_l}, \quad \boldsymbol{e}_{ij} \longmapsto \boldsymbol{x}_{ij}, \quad X = \{\boldsymbol{x}_{ij} \mid (i, j) \in \mathcal{E}\} \tag{22}$$

$$\epsilon_v : \mathbb{R}^{F_v} \longrightarrow \mathbb{R}^{F_l}, \quad \boldsymbol{v}_i \longmapsto \boldsymbol{x}_i, \quad \mathcal{X} = \{\boldsymbol{x}_i \mid i \in \mathcal{V}\} \tag{23}$$

The latent space dimension F_l is set as a hyperparameter to regulate its expressiveness, as well as way to control the number of trainable parameters that define the algorithm.

2.3.2 Processor

The processor plays a central role in the algorithm, as it facilitates the exchange of nodal information between vertices through message passing, transmitting the flow of information between the actuators and the central mesh through the latent space. When subdividing each step of the processor, two distinct stages are defined. In the first stage, information from the nodes is gathered in conjunction with the edges embeddings, $\boldsymbol{x}_{ij}$ and integrated into a MLP, applied to both the edges of the central graph, f^m and those of the actuator graph, f^w. In the second stage, the latent representation of the edges is incorporated through a permutation-invariant aggregation function, ϕ, combining it with the node embedding in another MLP, f^v to obtain the latent representation of the vertices. This process is repeated for a number of hyperparameterized steps, adding the latent representations in a

residual manner during the loop [15],

$$f^m(\boldsymbol{x}_{ij}^m, \boldsymbol{x}_i^m, \boldsymbol{x}_j^m) \to \boldsymbol{x}_{ij}^{\prime m}, \quad f^w(\boldsymbol{x}_{ij}^w, \boldsymbol{x}_i^w, \boldsymbol{x}_j^w) \to \boldsymbol{x}_{ij}^{\prime w},$$
$$f^v\left(\boldsymbol{x}_i, \phi(\boldsymbol{x}_{ij}^{\prime m}), \phi(\boldsymbol{x}_{ij}^{\prime w})\right) \to \boldsymbol{x}_i^{\prime} \tag{24}$$

where $(\cdot, \cdot, \cdot)$ denotes vector concatenation and $\boldsymbol{x}_i^{\prime}$ and $\boldsymbol{x}_{ij}^{\prime}$ are the updated nodal and edge latent representations.

2.3.3 Nodal vanilla decoder

For the vanilla data-driven framework the last block extracts the total derivative, $\dot{\boldsymbol{z}}$, directly from the latent vectors of the mesh nodes with a single decoder δ, therefore defining the temporal evolution of the model $\dot{\boldsymbol{z}}_i \in \mathbb{R}^{F_z}$, where F_z denotes the dimension of the state variables, as seen in Fig. 3,

$$\delta : \mathbb{R}^{F_l} \longrightarrow \mathbb{R}^{F_z}, \quad \boldsymbol{x}_i^{\prime m} \longmapsto \dot{\boldsymbol{z}}_i, \quad \mathcal{X}^{\prime} = \{\boldsymbol{x}_i^{\prime} \mid i \in \mathcal{V}\}. \tag{25}$$

2.3.4 Nodal physics-informed decoder

For the hybrid framework the last block extracts as well the relevant physical output information of the system from the node latent feature vector $\boldsymbol{x}_i^{\prime} \in \mathbb{R}^{F_{y,v}}$ and from the edge latent feature vector $\boldsymbol{x}_{ij}^{\prime} \in \mathbb{R}^{F_{y,e}}$, implemented with two decoders built by independent MLPs (δ_v, δ_e),

$$\delta_v : \mathbb{R}^{F_l} \longrightarrow \mathbb{R}^{F_{y,v}}, \quad \boldsymbol{x}_i^{\prime m} \longmapsto \left(\frac{\partial E}{\partial z}, \frac{\partial S}{\partial z}\right), \tag{26}$$

$$\delta_e : \mathbb{R}^{F_l} \longrightarrow \mathbb{R}^{F_{y,e}}, \quad \boldsymbol{x}_{ij}^{\prime m} \longmapsto (\boldsymbol{l}, \boldsymbol{m}). \tag{27}$$

In each case, the subindex v corresponds to the nodal decoding of the energy gradient $\frac{\partial E}{\partial z}$ and the entropy gradient $\frac{\partial S}{\partial z}$. The subindex e, on the other hand, refers to the decoding process of the flattened operators $\boldsymbol{l}$ and $\boldsymbol{m}$ from each edge, also considering the self-loop.

The extraction of all these components, instead of directly extracting the gradients, has a primary purpose: the imposition of a metriplectic structure that restricts the network to
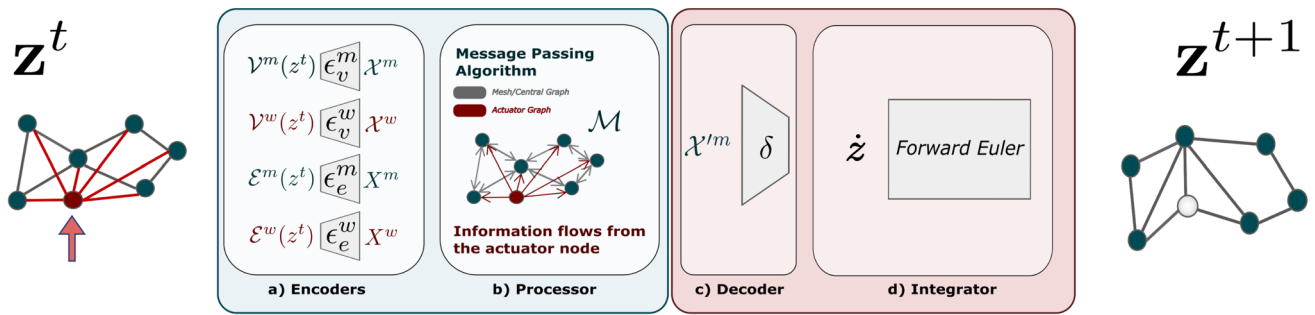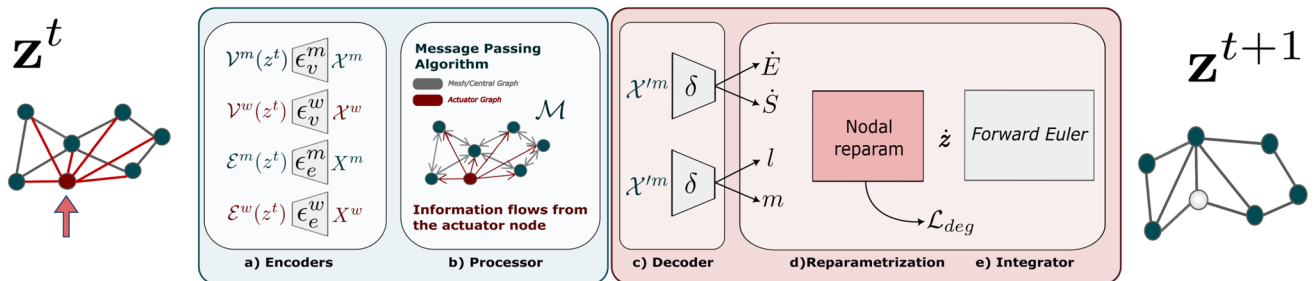
Fig. 3 Overview of the algorithm block scheme for predicting single-step state variable changes in the vanilla graph framework. **a** The encoder transforms the node and edge features into the latent space. **b** The processor distributes information across the graph using $\mathcal{M}$ message-passing modules. **c** The decoder extracts the gradient vectors from the processed graph. **d** The integrator predicts the state variables for the next time step. This entire sequence is repeated iteratively to generate the system's dynamic rollout

Fig. 4 Overview of the algorithm block scheme for predicting single-step state variable changes in the hybrid graph framework. **a** The encoder transforms the node and edge features into the latent space. **b** The processor distributes information across the graph using $\mathcal{M}$ message-passing modules, which we will refer to as simply "the number of passes" hereafter. **c** The decoder extracts the energy and entropy gradients from each node in addition to the flattened operators from each edge. **d** The reparametrization aggregates the boundary terms as seen in Eq. (17) to each node subsystem, predicting a gradient vector based on the GENERIC formulation **e** The integrator predicts the state variables for the next time step. This entire sequence is repeated iteratively to generate the system's dynamic rollout

solutions that geometrically satisfy the thermodynamics of the system, with the added objective of imparting a physical bias to the network to enhance its robustness in purely data-driven scenarios. Figure 4 illustrates this implementation through the metriplectic imposition, incorporating hard constraints for constructing linear operators alongside soft constraints applied to the loss function.

2.4 Learning procedure and data properties

The master database contains 190 simulations across five distinct anatomies [39], coined as $\{L1, \ldots, L5\}$ (Fig. 5 and Table 1). They feature mesh sizes ranging from 460 to 680 nodes per mesh. Although the dimensions varied, for the different anatomies, they were approximately $12 \times 10 \times 10$ cm (Width x Height x Thickness). We assume Dirichlet boundary conditions on the visceral face (visually displayed in Fig. 6). Data normalization entails standardizing the state tensor, which is represented as a 12-element vector per node for each snapshot and simulation. This vector comprises the 3 positional components (in m), 3 velocity components (in

m/s), and 6 stress values (in Pa), ensuring consistent scaling and unit uniformity across the dataset. This vector includes three positional dimensions, three velocity dimensions, and six components of the Cauchy stress tensor.

Of the five geometries, four are used for training, resulting in 760 simulations, each with 20 time steps. For each simulation, the number of imposed displacements (between 1 and 3) was randomly selected. Each displacement was modeled as an imposed node displacement, at a randomly selected node. The displacement vector was applied in a direction normal to the liver surface (for simplicity) with randomly selected sense - inwards (pushing the liver), outwards (pulling the liver). The magnitude was randomly selected between 0.5 cm and 2.5 cm.

For validation, 20% of the simulations from the remaining geometry ($L5$) are used. The primary testing dataset is based on the 190 total simulations of $L5$, including those also used for validation to avoid biases.

The frameworks—both vanilla and hybrid—are designed with approximately 240,000 trainable parameters. The training process utilizes the Adam optimizer along with a learning
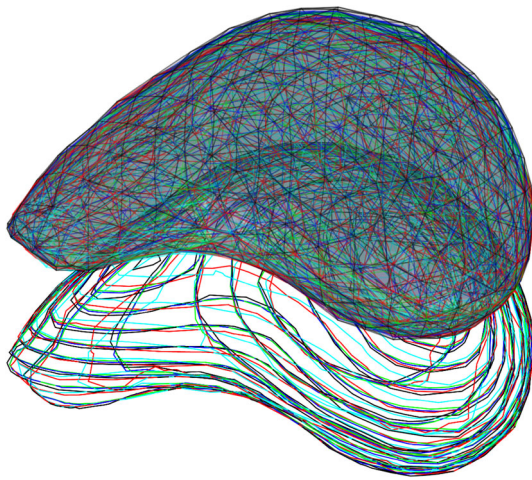
Fig. 5 Overlay of the different hepatic geometries used, along with a projection of the representative contour curves. Every color is associated with a different anatomy, providing a visual representation of the anatomical variability inherent to the database. Image top - meshes. Image bottom - level set curves for each anatomy

rate scheduler that reduces the rate once a plateau in training loss is reached. Model evaluation includes five standard models and one fast inference model.

Training takes approximately two days on an RTX 3090, while weight selection for test inference is done through early stopping based on the mean relative rollout error across the complete test dataset.

Both methodologies are formulated as minimization problems for a given loss function, $\mathcal{L}$. In the case of the vanilla model $\mathcal{L}_{\text{vanilla}}$, the cost function only evaluates the data loss $\mathcal{L}_{\text{data}}$ representing as the MSE along the graph nodes and state variables between the predicted and the ground-truth time derivative of the state vector in a given batch of snapshots,

$$\mathcal{L}_{\text{vanilla}} = \mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=0}^{N} \frac{1}{D} \left\| \frac{dz_i^{\text{GT}}}{dt} - \frac{dz_i^{\text{net}}}{dt} \right\|_2^2, \quad (28)$$

being N the total number of nodes per batch and D the total number of features.

The hybrid model loss $\mathcal{L}_{\text{hybrid}}$, in contrast, incorporates additional loss terms that account the residual of the degeneracy conditions, Eqs. (20) and (21),

$$\mathcal{L}_{\text{deg}} = \frac{1}{N} \sum_{i=0}^{N} \left\| L \frac{\partial S}{\partial z_i} + M \frac{\partial E}{\partial z_i} \right\|_2^2, \quad (29)$$

modulated by the parameter λ_d to control the influence of the physical term,

$$\mathcal{L}_{\text{hybrid}} = \mathcal{L}_{\text{data}} + \lambda_d \mathcal{L}_{\text{deg}}. \quad (30)$$

Table 1 Specific data for each anatomy and, consequently, for the meshes resulting from its spatial discretization

Geometry	Volume [m^3]	Nodes	Elements
L1	2.71E–4	530	1734
L2	2.33E–4	460	1449
L3	2.14E–4	492	1547
L4	2.58E–4	680	2299
L5	2.61E–4	665	2203

The training subset covers from **L1** to **L4**, while **L5** is reserved for testing on a geometry unknown to the network

The value of λ_d is set to 5 for all trials, as sensitivity analyses have determined that its optimal range lies between 1 and 20.

2.4.1 Material properties

To determine the $F(z^t)$-function, synthetic data will be employed. These will be obtained by high-fidelity finite element simulations assuming a homogeneous simplification of liver tissue, omitting vascularization and other relevant anatomical features. The hyperelastic part of the response response will be defined by a strain energy density function W.

To model this hyperelastic behavior, the Ogden model [40] will be used,

$$W = \sum_{i=1}^{N} \frac{\mu_i}{\alpha_i} (\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3) + \sum_{i=1}^{N} \frac{1}{d_i} (J-1)^{2i}, \quad (31)$$

by assuming $N = 3$, where:

- W: the strain energy density function for the hyperelastic material.
- μ_i: material parameters that define the contribution of each mode to the energy density.
- α_i: exponents that characterize the non-linear behavior of the material.
- $\lambda_1, \lambda_2, \lambda_3$: the principal stretches of the material.
- J: the determinant of the deformation gradient, representing the volumetric change of the material.
- d_i: parameters related to the volumetric response, influencing the contribution of the volumetric term.
- N: the number of terms in the summation, allowing for a more detailed representation of the material's response.

Alongside, the viscous part of the behaviour will be modeled through Prony series, which are suitable for capturing the time-dependent behavior of the material [41]. The Prony

series representation is given by

$$G(t) = G_0 \left(\alpha_\infty + \sum_{k=1}^{N} \alpha_k e^{-t/\tau_k} \right), \qquad (32)$$

where:

- $G(t)$: the dynamic shear modulus as a function of time in pascals.
- G_0: the initial shear modulus (the value at $t = 0$) in pascals.
- α_∞: represents the long-term behavior of the material (the steady-state value).
- α_k: coefficients that weight the contribution of each relaxation mode.
- τ_k: relaxation times corresponding to each mode, dictating how quickly the material responds over time.
- N: the number of terms used in the series, allowing for a more accurate representation of the material's behavior.

Finally, the material parameters will be calibrated based on existing experimental literature to ensure the model accurately reflects the properties of hepatic tissue, with parameter values detailed in Table 2 [8]. For precise simulations, the liver anatomy is discretized using finite elements, as shown in Fig. 6, which highlights the meshing strategy and the geometric detail captured to represent the hepatic structure.

3 Results

In this section, we present and compare the results of the two approaches considered: the vanilla graph network, assumed as the baseline, and the hybrid graph neural network incorporating the metriplectic structure. To ensure a fair comparison, both methods are tuned to have an equal number of parameters. Additionally, each model is trained using different seeds

Table 2 Parameters of the Ogden model and Prony series constants [8]

Model	Parameter	Value
Ogden	E (Pa)	12,598
	μ_L (Pa)	364.74
	α_L	16.19
	d (Pa^{-1})	9.525×10^{-5}
Prony Series	α_1	0.56057
	τ_1 (s)	6.36173
	α_2	0.40585
	τ_2 (s)	0.73348

to ensure reproducibility. All evaluations are also conducted on a dataset of 190 simulations derived from an unseen anatomy, referred to as *Test* from this point forward.

3.1 Evaluation metrics

The Root Mean Square Error (RMSE) values

$$\mathrm{RMSE}(z_{\mathrm{ref}}, z_{\mathrm{pred}}) = \sqrt{\frac{1}{n_{\mathrm{snp}}} \sum_{i=1}^{n_{\mathrm{snp}}} \left(\frac{1}{n_v} \sum_{j=1}^{n_v} \frac{\|z_{\mathrm{ref}} - z_{\mathrm{pred}}\|_2^2}{D} \right)},$$
$$(33)$$

in Table 3 show that the hybrid model outperforms the vanilla model across all considered state variables. Specifically, the hybrid approach achieves lower RMSE values for q, v, and σ, suggesting that the incorporation of the metriplectic structure improves the model's predictive accuracy.

Let n_v represent the total number of nodes per mesh, and n_{snp} the number of snapshots taken per simulation. In this case, all snapshots from each *Test* subset are considered collectively.
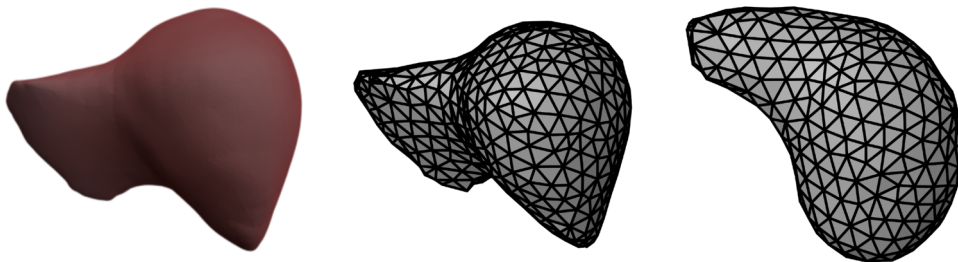

Fig. 6 Finite element discretization of the liver. This figure demonstrates the meshing strategy designed to accurately capture the geometric and structural intricacies of hepatic tissue, providing a reliable foundation for computational simulations. The mesh is composed of 4-node linear tetrahedral elements. The first two illustrations show the visceral face of the liver: the rendering of the liver parenchyma (left) and the corresponding mesh for this perspective (center). The third illustration provides the equivalent mesh representation of the liver's anterior face (right)

Table 3 RMSE comparison between three Hybrid and Vanilla Models across all snapshots in the testing subset, evaluated over five passes and approximately 200,000 trainable parameters

	Hybrid	Vanilla
q (m)	$(3.730 \pm 0.06) \times 10^{-4}$	$(4.060 \pm 0.11) \times 10^{-4}$
v (m/s)	$(7.430 \pm 0.06) \times 10^{-4}$	$(8.430 \pm 0.34) \times 10^{-4}$
σ (Pa)	$(1.664 \pm 0.009) \times 10^{3}$	$(1.755 \pm 0.037) \times 10^{3}$

The *Test* subset includes 190 untrained simulations in an entirely new anatomy, coined as **L5**.

To further assess the models' performance, we compute the Relative Root Mean Square Error (RRMSE),

$$\text{RRMSE}(z_{\text{ref}}, z_{\text{pred}}) = \sqrt{\frac{1}{n_{\text{snp}}} \sum_{i=1}^{n_{\text{snp}}} \left(\frac{1}{n_v} \sum_{j=1}^{n_v} \frac{\|z_{\text{ref}} - z_{\text{pred}}\|_2^2}{\|z_{\text{ref}}\|_\infty^2} \right)}. \tag{34}$$

Similar to the RMSE results, the hybrid model demonstrates superior performance by consistently producing lower RRMSE values across all state variables. The statistical analysis reported in Table 4 shows the importance of these gains. To extend the discussion, other models with longer-range message passing procedures are explored in Appendix B.

3.2 Inference performance analysis for each simulation rollout

Absolute and relative metrics are excellent indicators for gaining a general overview of the model's performance, however, it is essential not to lose sight of the fact that one of the main objectives is to ensure adequate robustness during the full temporal evolution, avoiding significant impacts from the residual error accumulations generated by the recurrent inference on the model. First, the boxplots presented in Fig. 7 compare the distribution of relative error (RRMSE) for the three state variables in each simulation rollout.

These results correlate with the previously mentioned metrics, demonstrating that the hybrid methodology enhances both the predictive capacity and the robustness of the model, for architectures trained with five message passing steps as well as with one.

The global value is measured using the mean of all snapshots and all nodes, even when the initial snapshots exhibit lower magnitudes in the state variables. Additionally, despite averaging across all nodes, the local displacement only induces stress and deformation peaks in a small subset of them. Although the difference can be observed in terms of relative or absolute error, it is important to note that this

is asessed at the full graph rollout level, making the predictive improvement associated with error reduction difficult to interpret using only global metrics. The deepest statistics defined in Table 4 along with Figs. 8 and 9 support the main argument, illustrating the effect of physics bias in a simulation with an untrained geometry.

The main hypothesis suggests that the metriplectic structure, which enforces both weak and strong constraints, improves the stability of nodal inference and prevents stress-induced error spikes from negatively impacting positional prediction. This aligns with the observation that significant error accumulation tends to occur around areas of imposed displacement and near stress concentrators.

The maximum response times exceed 600 Hz for one-step models, while the average times for the more refined models hover around 150 Hz for both methodologies. However, a more detailed description of the evolution of the response time with an increased pass count in the graph processor can be found in the Appendix C.

3.3 Robustness against previously unseen anatomies

Another argument that complements the high feedback rates of the network is its robustness against untrained anatomies and meshes, making it highly suitable for precision medicine contexts using digital twins. To further evaluate this capability, we have supplemented the main test database with 33 new and untrained simulations within an already trained geometry, designating this subset as *Extra*. This subset serves as an intermediate point between the 190 *Train* sampled simulations and the 190 *Test* simulations, which are used to assess the rollout performance.

In Fig. 10 we compare the prediction accuracy across the three datasets, showing that, as a data-driven model, the more we extrapolate from the training space, the worse the inference will be. The best results from the rollout are consistently obtained with trained geometries and loads. In contrast, error accumulates progressively in models facing different load conditions and meshing; however, this growth does not spike for any of the three defined state variables. These findings align with the behaviors observed in the literature and highlight the robustness of graph networks, emphasizing their potential in practical precision medicine applications dependent on digital twins.

As an additional resource, Appendix E outlines the network performance across ten simulations conducted at four distinct load points for an untrained anatomy. These load conditions, which differ from those encountered during training, serve as a confirmation of the actuator's adaptability and robustness.

Fig. 7 Comparison of RRMSE rollouts between the five-pass and one-pass models across 190 unseen simulations in a new geometry. The blue boxes represent the error distribution for the vanilla model, while the red boxes show the distribution for the hybrid model
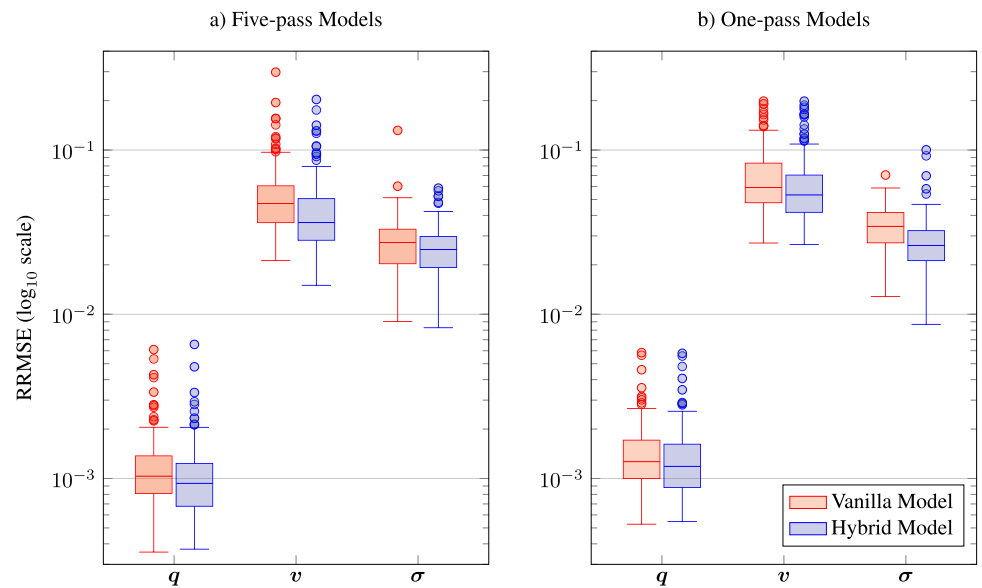

a) Five-pass Models

b) One-pass Models

Table 4 Statistical comparison of performance improvements between the hybrid model and the Vanilla Model across state variables q, v, and σ

	μ (%)	σ (%)	T-test (p-value)	Significant	Wilcoxon (p-value)	Significant
5 Passes						
q	11.70	6.41	1.50×10^{-6}	Yes	4.37×10^{-10}	Yes
v	18.51	14.17	5.34×10^{-10}	Yes	2.78×10^{-16}	Yes
σ	7.77	21.95	8.04×10^{-4}	Yes	4.60×10^{-9}	Yes
1 Pass						
q	6.19	−3.37	4.72×10^{-3}	Yes	3.43×10^{-6}	Yes
v	9.46	−0.66	1.96×10^{-4}	Yes	3.54×10^{-9}	Yes
σ	19.15	−16.52	5.09×10^{-23}	Yes	7.02×10^{-27}	Yes

The table presents the mean (μ) and standard deviation (σ) of percentage improvements, where μ represents the average improvement and σ quantifies the variability. Negative σ values indicate performance degradation rather than improvement. Statistical significance is assessed using paired t-tests and Wilcoxon signed-rank tests, with p-values reported. Results confirm that the GENERIC, hybrid model achieves statistically significant ($p < 0.05$) improvements over the vanilla model for all state variables under both 5-pass and 1-pass configurations, demonstrating its robustness and superior generalization capabilities

Ground Truth

Vanilla GNN

Hybrid GNN

$||\boldsymbol{u}||_2$ 0.0   0.00317   0.00951   0.0127
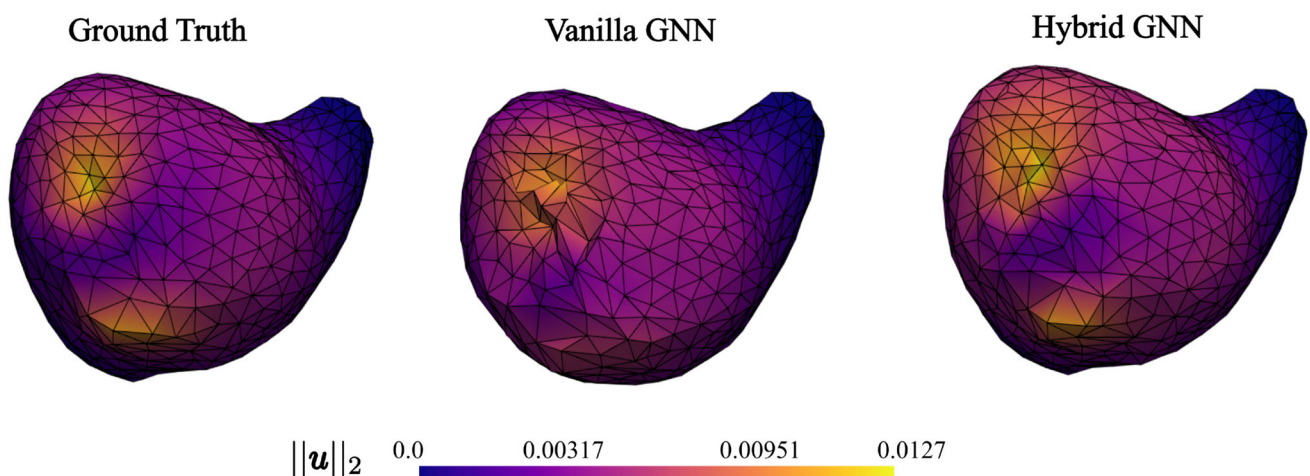
Fig. 8 Comparison of the final inference step between the two proposed models and the finite element solution, used as ground truth, for previously unseen geometries and load conditions. The color scale represents the 2-norm of the displacement vector on each node
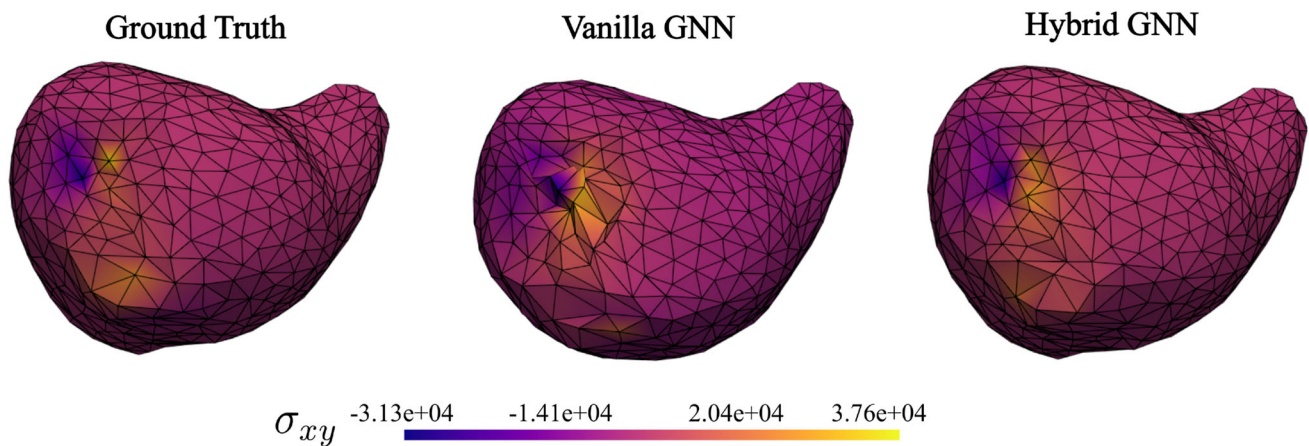
Fig. 9 Comparison of the final inference step between the two proposed models and the numerical solution, used as ground truth, for untrained geometries and unseen load conditions. The color scale represents the σ_{xy} (Pa) scalar value on each node
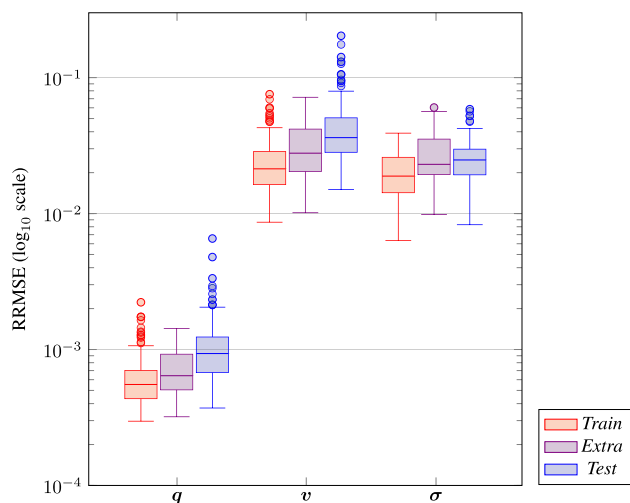

Fig. 10 Comparison of rollout Relative Root Mean Square Error (RRMSE) across three datasets: *Train* (trained data), *Extra* (unseen load states with known anatomy), and *Test* (untrained anatomies and load states, main testing dataset) for the parameters q, v, and σ

4 Conclusions

The proposed Graph Neural Network (GNN) framework exhibits significant potential for real-time simulations of clinical procedures that involve complex, multi-point manipulations of the liver. This computational paradigm is particularly well-suited for surgical planning, where accurate and efficient modeling of liver deformations under external interactions is crucial. The trained GNN models have been rigorously evaluated under previously unseen loading conditions, demonstrating robust generalization capabilities (Appendix E). Furthermore, the adaptability of the trained network extends to novel and previously unencountered geometries, including anatomical variations across different patients. This generalizability enables the deploy-

ment of a single base model, with minimal fine-tuning, for patient-specific simulations covering a broad spectrum of pathological conditions and surgical interventions. Additionally, integrating this technology with haptic feedback systems [42, 43] could further expand its applicability to medical training, allowing surgeons to refine their proficiency in novel procedures and patient-specific scenarios.

In regard to data-driven neural networks for supervised learning, inference is inevitably subject to some degree of bias. Balancing the amount of data needed to ensure model robustness and accuracy in extrapolation remains a complex challenge. However, hybrid methodologies have proven to be more effective than purely data-driven models, particularly in environments with sparse or limited data [24, 28, 44]. By incorporating soft constraints during loss optimization, imposing mathematical structures on the decoding process, or using base distributions to enhance the expressiveness of the latent space-as demonstrated in variational autoencoders [45]-we hypothesize that neural networks can achieve more robust inference.

Validating this idea through computational mechanics, we teach the network to understand the physical behavior of soft tissue and act as a temporal integrator. However, we face the challenge that recurrent inference during the rollout tends to accumulate error. Despite this, the network's understanding of the underlying physics, augmented by data and noise (discussed in more detail in Appendix A), aids in achieving realistic dynamic behavior. Additionally, graph neural networks exhibit impressive extrapolation capabilities when applied to unfamiliar geometries or meshes in finite element simulations, suggesting the model's ability to grasp energy exchanges between the actuator and the receiver.

Therefore, integrating real-time simulation with precision medicine through deep learning is feasible, paving the way for the medical use of patient-specific digital twins. However,

new solutions bring new challenges. In this case, the interpretability of message passing and scalability issues with larger meshes remain as the primary limitations in graph architectures. Due to the main mesh size limitation, we cannot generate a finer mesh, so we work with approximately 1000 nodes consistently. This allows us to train and test faster while exploring the model's performance.

These limitations may be addressed by exploring the impact of message passing on model learning while integrating graph representations with model order reduction or multiresolution techniques [46–48] to enhance performance and reduce inference times. On the other hand, the interpretability of physics-based biases is also a fundamental requirement for the acceptance of digital twins, specially for critical applications. This holds true for industrial application scenarios where safety and accuracy are paramount (industrial monitoring, aircraft maintenance, etc.), as well as for medical applications in personalized medicine pipelines (virtual surgery planning, telemedicine diagnostics, etc.). However, interpretability requirements should not hinder or constrain a much needed robustness to unknown differential behaviors.

On a completely different setting, we would like to address some words to the problem of data acquisition. One of the most common ways of developing a digital twin is to do it on the basis of measurements taken on videos. On the one hand, this implies being able to take data at a rate of about 25–60 frames per second and, on the other hand, being able to simulate at the same rate. Our technique has proven to be able to do the latter with the means available to us today.

The first part, however (data collection) presents significant challenges. Perhaps the most important is that certain variables are not measurable by video. This is particularly the case for stresses. How to make a digital twin then, if we are not going to be able to measure stresses, and displacements and velocities can only be obtained on the surface of the solid, as it is opaque, and possibly in a partial view (this is usually the case in cholecystectomy operations, for example)?

Our group has already devoted considerable attention to this problem in the recent past, see [49]. An active learning technique has been developed which has allowed it to adapt to such limiting circumstances, showing remarkable levels of accuracy.

A specific aspect of biomechanical applications remains however to be analysed: registration (alignment between the real and synthetic organ) for measurement and eventual use in augmented reality applications, for example, is particularly difficult. In addition to the partial view of the organ (if we are talking about the liver), the highly reflective surface of the organ causes glare, which hinders the tracking of deformations.

In any case, this does not constitute, in our opinion, an insurmountable difficulty, and we will devote our efforts to it in the near future.

Therefore, we can conclude that this work provides a consistent roadmap for further reducing the heavy data dependency of supervised deep learning approaches. The improvements achieved through the implementation of hybrid-AI have enabled the training of fast, reliable liver digital twins, paving the way for real-time haptic surgical simulations with a high degree of accuracy.

A Noise data augmentation

The rollout's inference can be defined as the recursive prediction of the model's temporal evolution, where accumulated error typically increases with successive predictions, affecting new outputs of the deep neural network. For this reason, robustness mechanisms are essential to prevent error spikes that could destabilize the model.

In this study, the deliberate inclusion of noise is validated as a tool for data augmentation rather than for filtering/denoising purposes. This concept is established by defining z_i^t as the state of a node at a specific time step, represented by the state vectors $\boldsymbol{q}_i$, $\boldsymbol{v}_i$, and $\boldsymbol{\sigma}_i$,

$$z_i^t = \begin{pmatrix} \boldsymbol{q}_i \\ \boldsymbol{v}_i \\ \boldsymbol{\sigma}_i \end{pmatrix}, \tag{35}$$

where the noise methodology defines $z_i'^t$ as the node representation with added noise. Specifically, for the position vector $\boldsymbol{q}_i$, a stochastic term proportional to the mean of the edge distances $\overline{\boldsymbol{u}}_{ij}$ is included,

$$\boldsymbol{q}_i' = \boldsymbol{q}_i + \overline{\boldsymbol{u}}_{ij}\epsilon_q n_q \quad \epsilon_q \sim \mathcal{U}(-1, 1), \tag{36}$$

capturing systematic positional variations within the discretization range and scaled with an hyperparameter n_q.

For the velocity vector $\boldsymbol{v}_i$ and the stress vector $\boldsymbol{\sigma}_i$, noise is introduced as a relative perturbation, scaled by a factor n_v,

$$\boldsymbol{v}_i' = \boldsymbol{v}_i + \boldsymbol{v}_i\epsilon_v n_v, \quad \epsilon_v \sim \mathcal{U}(-1, 1), \tag{37}$$

$$\boldsymbol{\sigma}_i' = \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_i\epsilon_\sigma n_v, \quad \epsilon_\sigma \sim \mathcal{U}(-1, 1). \tag{38}$$

These stochastic terms are also uniformly distributed, with distribution denoted as $\mathcal{U}$, within the interval $[-1, 1]$. Terefore, z_i' can be defined as,

$$z_i'^t((z_i, \overline{\boldsymbol{u}}_{ij}, n_q, n_v)) = \begin{pmatrix} \boldsymbol{q}_i' \\ \boldsymbol{v}_i' \\ \boldsymbol{\sigma}_i' \end{pmatrix} = \begin{pmatrix} \boldsymbol{q}_i + \overline{\boldsymbol{u}}_{ij}\epsilon_q n_q \\ \boldsymbol{v}_i + \boldsymbol{v}_i\epsilon_v n_v \\ \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_i\epsilon_\sigma n_v \end{pmatrix}, \tag{39}$$

which consequently generates that $\frac{dz_i}{dt}$ denotes the temporal variation of the state variables, but taking into account the variation aggregated by the noise,

$$\frac{dz_i^{\mathrm{GT}}}{dt} = \frac{z_i^{t+1} - z_i'^t}{dt}, \tag{40}$$

so that naming the network output as $\frac{dz_i^{\mathrm{net}}}{dt}$, the loss function for the data and consequently the model optimization will be given by Eq. (28). The opposite approach, denoising, would have been not to consider this noise variation in the cost optimization, but only at the network input such that,

$$\frac{dz_i^{\mathrm{GT}}}{dt} = \frac{z_i^{t+1} - z_i^t}{dt}. \tag{41}$$

This approach establishes a learning process in which the network learns to construct the integrator from positions that are not always exact, building essential robustness for the recurrent inference during rollout.

Additionally, for this project we have not optimized a deep sweep of hyperparameters to regulate the noise in the three state vectors. However, we have worked with some noise values that have proven to improve the inference, but without being able to ensure that they are optimal. These values are 10^{-4} for n_q and 10^{-3} for n_v.

B Model scalability and message passing efficiency

As we have observed, increasing the steps in the graph processor scales almost linearly for both training and inference. However, even with the proper distribution of information provided by the actuator graph, extending each node's attention to its surroundings significantly reduces error, as demonstrated in Fig. 11, especially when considering the previously presented error in Fig. 7. This effect is not only amplified in the inference on the main *Test* database, but it is also significantly correlated in the rollout error on the other datasets (*Train* and *Extra*, Fig. 12).

This main hypothesis is based on how each node infers a response that is mathematically dependent on the behavior of further nodes, which provides greater stability to the system without being significantly affected by an increase in the dimensionality of the latent space. While a higher-dimensional latent space aids in better projecting information related to the state variables, the weight of the inference seems to depend more on the proper optimization of message passing.
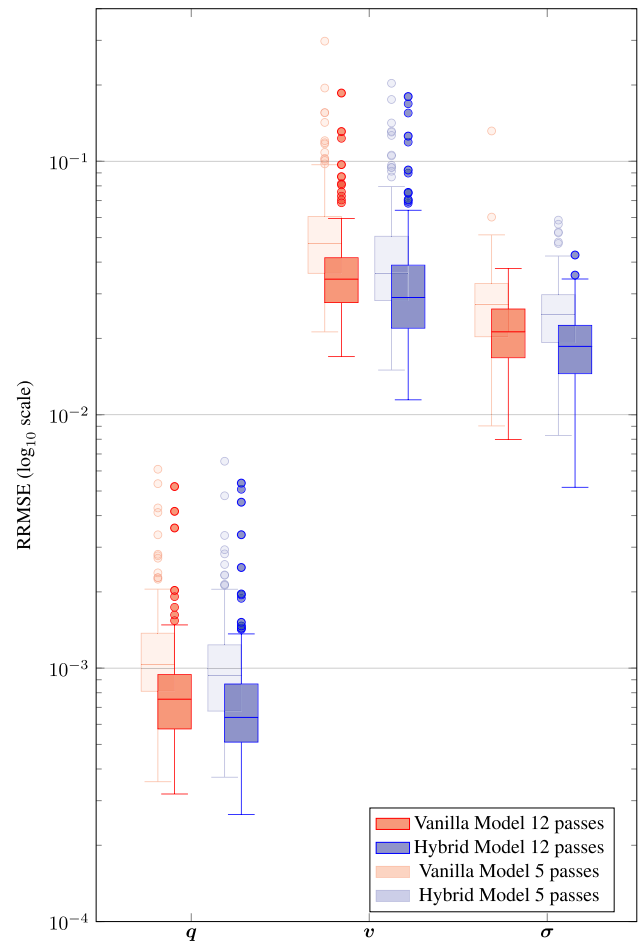

Fig. 11 Comparison of RRMSE rollouts between the five-pass and twelve-pass model across 190 unseen simulations in a new anatomy. The blue boxes represent the error distribution for the vanilla model, while the red boxes show the distribution for the hybrid model. The boxes with reduced opacity illustrate the error distributions previously displayed in the Fig. 7, serving as a visual aid for comparison

Accordingly, in Fig. 13, we observe that for the same test simulation, the model with a greater number of passes—where each pass (or hop) corresponds to one round of the aggregation process—better captures the locality of the deformation. This prevents its propagation and demonstrates a deeper understanding of the soft tissue properties.

As supplementary material, the temporal evolution of the twelve state variables is also presented in four different subplots, shown in Figs. 14 and 15. These illustrate the network's adjustment for inferring a node in an untrained geometry and load state. This analysis clearly validates how recurrent inference accumulates errors due to temporal integration. However, thanks to the appropriately biased methodology, along with the robustness achieved through data augmentation via noise, the simulation is prevented from diverging into significantly larger errors.

Fig. 12 Comparison of RRMSE rollouts between the five-pass and twelve-pass hybrid models across different datasets: *Train* (trained data), *Extra* (unseen load states with known geometry), and *Test* (untrained geometries and load states, main testing dataset) for the parameters q, v, and σ. The boxes with reduced opacity represent the error distributions from the five-pass results previously shown in the Fig. 10, serving as a visual aid for comparison
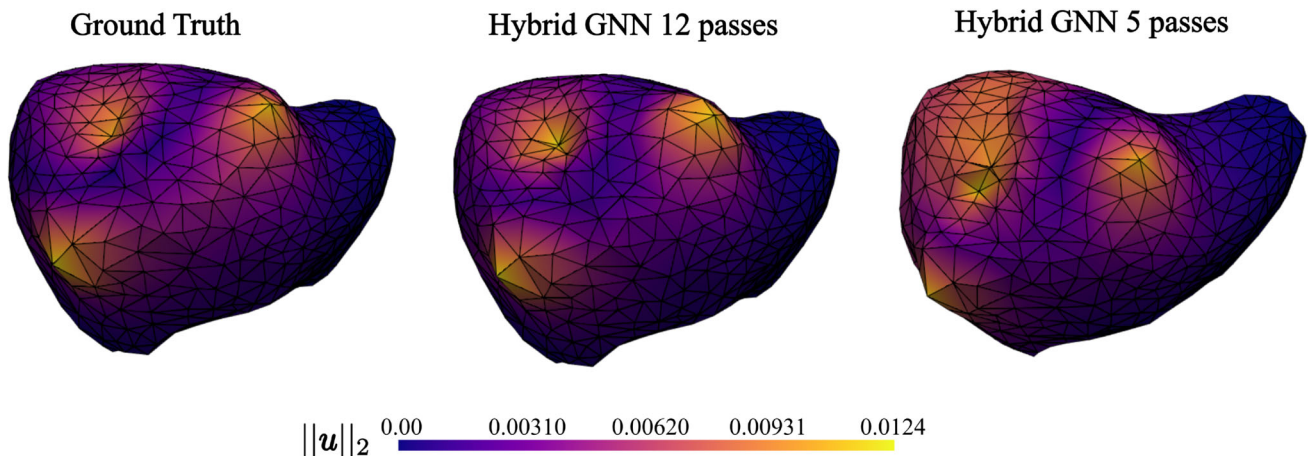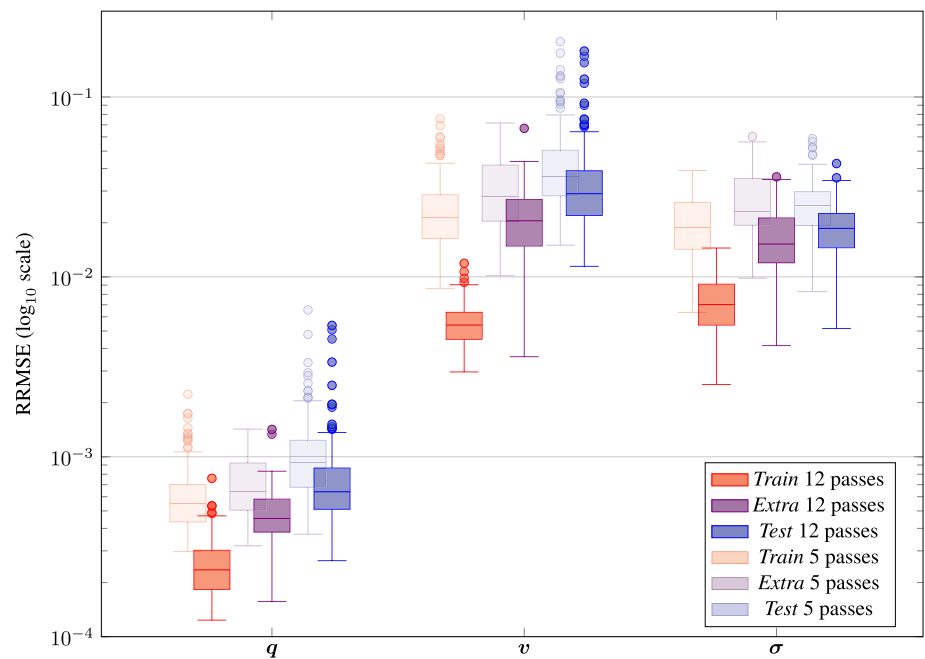

Fig. 13 Comparison of the final inference step between the two proposed models and the numerical solution, used as ground truth, for untrained geometries and unseen load conditions. The color scale represents the 2-norm of the displacement vector on each node

C Time response evolution with passing augmentation

In Graph Neural Networks (GNNs), model performance is strongly influenced by the flow of messages, which is controlled by the number of hyperparameter-tuned passes through the graph processing layers. When information is distributed and the geometric function depends on a broader attention area, inference tends to be more stable. However, this improvement comes at the cost of increased computational cost for both backpropagation and inference, as it is a non-parallelizable process. Figure 16 illustrates the evolution of response time as a function of the number of steps on a Nvidia RTX4090, for both the hybrid model and the vanilla model compared to the estimated computational cost of a conventional finite element solver, Table 5.

A reduction of several orders of magnitude is achieved compared to traditional numerical methodologies, reaching performance levels in the kiloHertz range in the most speed-optimized cases. This validates its utility in haptic environments and demonstrates its applicability even in scenarios with a high number of message iterations. It is also worth noting that this performance improvement across passes primarily scales during backpropagation, resulting in significantly higher training times. Even considering that the vanilla model is somewhat more efficient by not having to iterate the metriplectic integrator, at best, up to 168,7 speedup is achieved for the entire rollout.
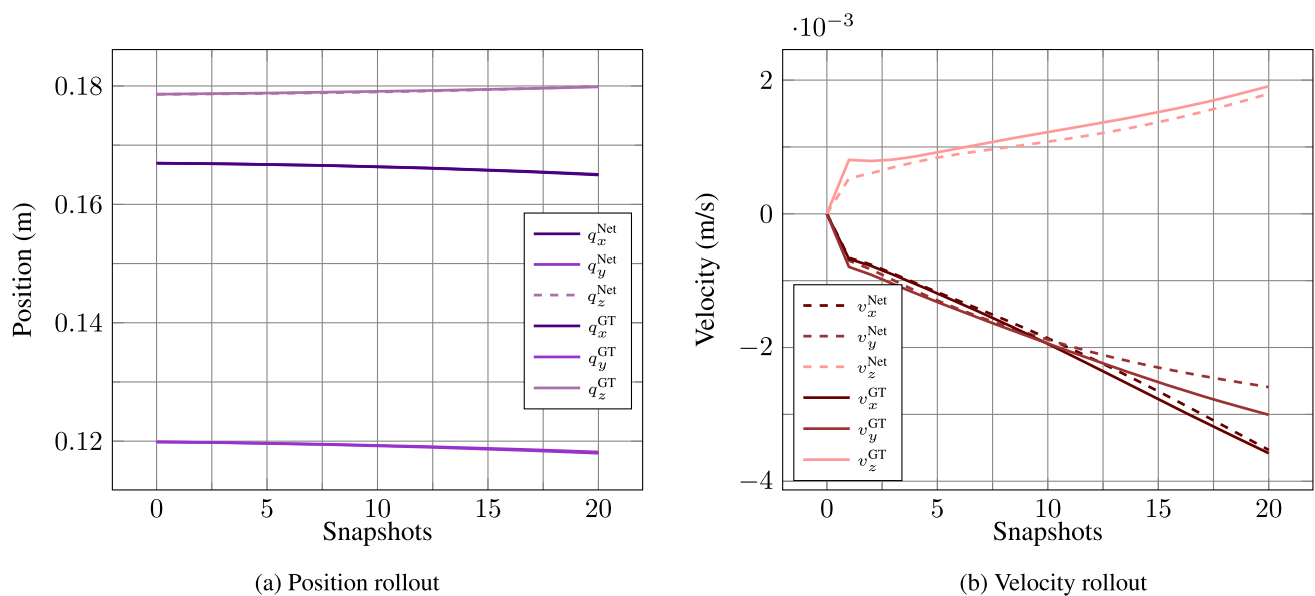
(a) Position rollout

(b) Velocity rollout

Fig. 14 Comparison of the rollout for first six state variables at a randomly selected node within one of the test inferences for the twelve-pass hybrid model. The dotted lines represent the network's recurrent prediction compared to the ground truth from numerical simulation, illustrating the accumulation of relative error over the course of temporal evolution
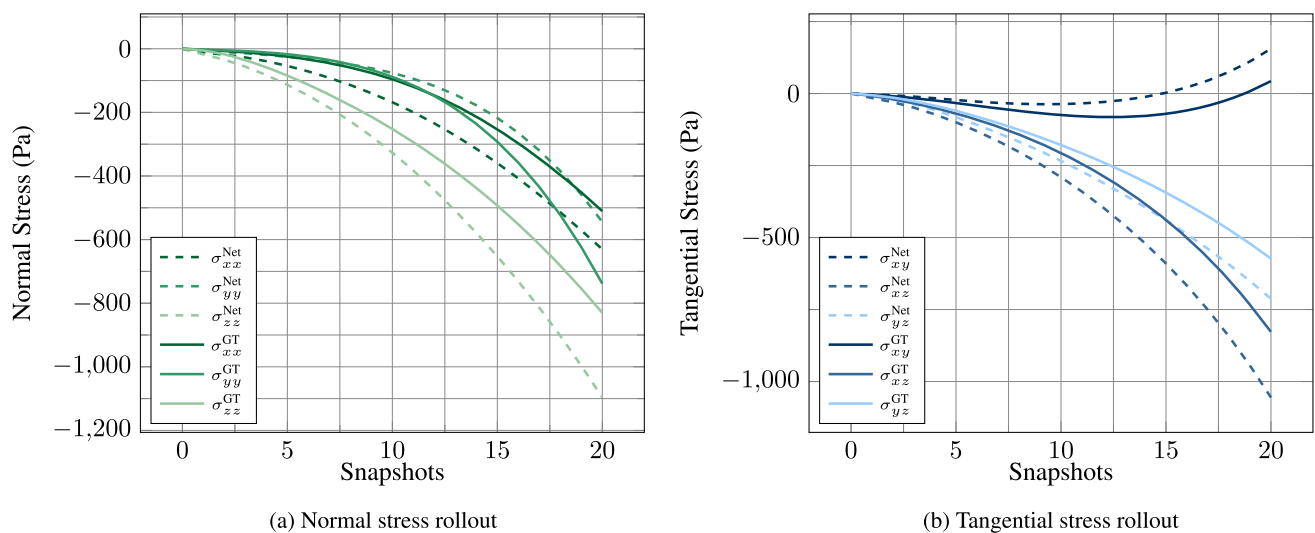

(a) Normal stress rollout

(b) Tangential stress rollout

Fig. 15 Comparison of the rollout for last six state variables at a randomly selected node within one of the test inferences for the twelve-pass hybrid model. The dotted lines represent the network's recurrent prediction compared to the ground truth from numerical simulation, illustrating the accumulation of relative error over the course of temporal evolution

D Multi-graph contribution to message passing

In the geometric context of graph networks, the discussion focuses on how message transmission —determined by the number of aggregation layers in the graph processor— positively influences the local understanding of the system's mechanical behavior. However, we must take into account that broader message distribution incurs a high computa-tional cost for both inference and backpropagation, reopening the debate between performance and efficiency.

Single-graph models [24, 28] suffer significantly from this limitation: if a local load is communicated relatively far away (at higher geodesic distances [50]), the message flow will not inform more distant nodes, a limitation that becomes more pronounced with finer spatial discretization. In this context, the proposed multigraph scheme [15]-validated in Appendix D-has a clear objective: to improve information distribution.
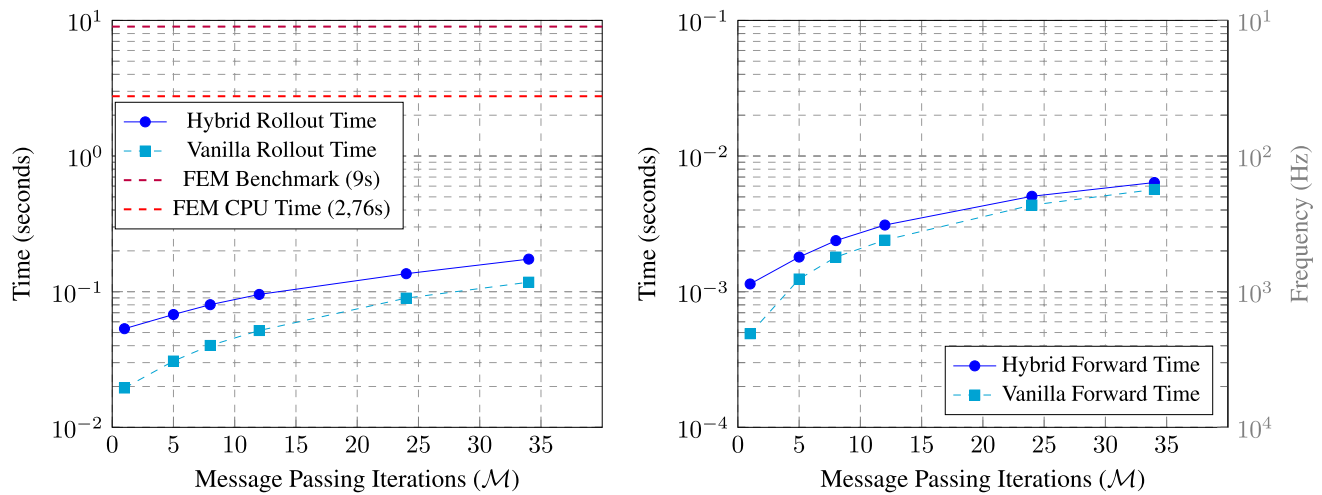
Fig. 16 Comparison of mean rollout and forward time performance between the Hybrid and Vanilla models during inference, evaluated for varying numbers of hyperparameterized passes in the processor $\mathcal{M}$. Additionally, FEM benchmark times are aggregated to provide a visual comparison

Table 5 Mean and standard deviation of the total FEM simulation time and CPU time on an Intel(R) Core(TM) i7-8700K CPU using a commercial finite element solver for the 5 digital hepatic twins

Simulation time	mean±SD (s)
Total time	9.00±1.41
CPU time	2.76±0.59

From the initial steps, it conveys not only where energy is introduced into the system but also the relative distance from each node.

For this project, the single-graph model appends the actuator information u_i node-wise during message transmission, which limits its influence on nodes farther from the radius defined by the message-passing modules hyperparameter $\mathcal{M}$,

$$f^m(\boldsymbol{x}_{ij}^m, \boldsymbol{x}_i^m, \boldsymbol{x}_j^m) \to \boldsymbol{x}_{ij}'^m, \quad f^v\left(\boldsymbol{x}_i, \phi(\boldsymbol{x}_{ij}'^m), \bar{\boldsymbol{u}}_i)\right) \to \boldsymbol{x}_i', \tag{42}$$

where $(\cdot, \cdot, \cdot)$ denotes vector concatenation and x_i' and x_{ij}' are the updated nodal and edge latent representations for the mesh-graph.

Figure 17 compares the relative error between single-graph models and the multigraph model, supporting the hypothesis that enhanced information transmission improves inference accuracy in scenarios where message passing is limited in relation to the density of the mesh. This comparison is performed under identical parameter, training, and testing conditions, validating the choice to adopt a multigraph approach. The results indicate that this approach not only enhances model performance but may also improve the efficiency of message flow within the graph processor.

For models trained with only five passes Fig. 17a and b, the multigraph scheme significantly improves information flow. However, when the number of passes is sufficient to cover the entire spatial discretization, see Fig. 17c, the multigraph effect is less critical to the network's performance. This may indicate that after twelve steps, the 500-node graph reaches a stationary state of information distribution, in which all nodes are theoretically aware of the external excitation affecting the system, and the inference performance is not constrained by it.

E Performance under severe extrapolation

For the training database, we chose that the number of loaded nodes range from a minimum of one to a maximum of three. In practice, endoscopic surgery uses a maximum of two surgical tools (a third incision could be made for the camera).

This appendix provides an overview of the models' performance under extrapolation conditions, more challenging than those encountered during the initial tests. Meanwhile we aim to validate the robustness and adaptability of the multigraph networks approach proposed.

The test set will consist of 10 simulations with four imposed displacement (traction or compression) application points. This will result in a system comprising four actuator graphs and one central graph, subsequently denoted as Extra*.

Table 6 presents a comparison of the absolute and relative errors between the two methodologies, highlighting their comparable robustness. This similarity is likely attributed to the shared geometric bias of both models, which primarily equips the network with effective generalization capabilities.

Fig. 17 Relative error comparison of data-driven model performance for single-graph versus multi-graph frameworks within the processor, under equivalent training conditions. The left panel displays results for both models with a single pass transmission, the right panel shows results for five passes, and the bottom panel presents results for twelve passes
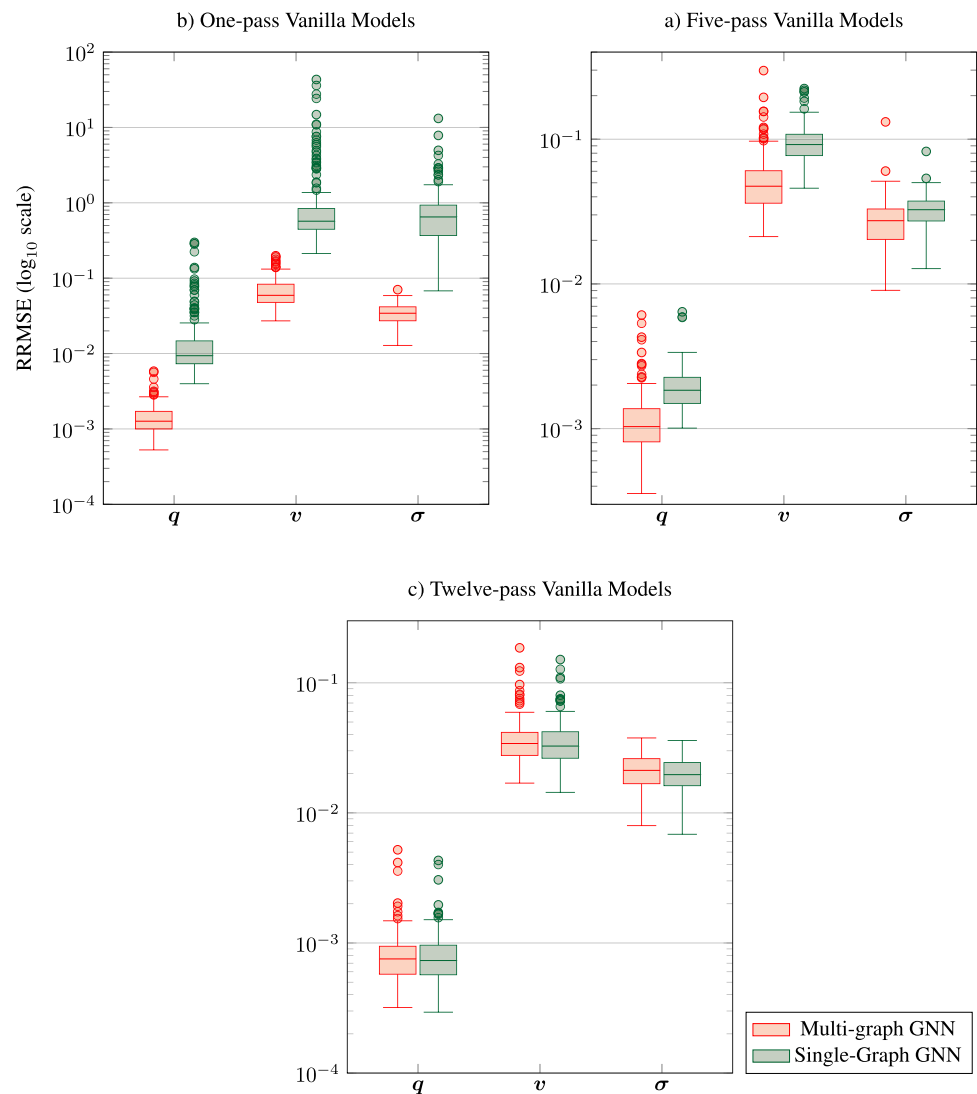

b) One-pass Vanilla Models

a) Five-pass Vanilla Models

c) Twelve-pass Vanilla Models

Multi-graph GNN
Single-Graph GNN

Table 6 Side-by-side comparison of RMSE and RRMSE across hybrid and vanilla models for all snapshots in the testing subset

	Hybrid	**Vanilla**
q (m)	3.620×10^{-4}	3.58×10^{-4}
v (m/s)	6.16×10^{-4}	6.23×10^{-4}
σ (Pa)	7.236×10^{2}	7.562×10^{2}
q	1.2717×10^{-3}	1.203×10^{-3}
v	5.13×10^{-2}	5.18×10^{-2}
σ	3.28×10^{-2}	3.40×10^{-2}

Each table evaluates five passes using approximately 2 million trainable parameters, with twelve passes applied to obtain the best-performing weights. The testing subset consists of 10 previously unseen simulations in the test anatomy, **L5**, featuring four loading points.

On the other hand, Fig. 18 illustrates the predictions for the final step of the rollout in two distinct simulations from the test set. These simulations correspond to a displacement field and a tangential stress field, respectively, highlighting the network's capabilities.

Finally, the boxplot presented in Fig. 19 compares the progression of the relative error during extrapolation to a more complex dataset against the primary test set (referred to as *Test*).

In order to also assess the extrapolation capability to geometries whose scale falls outside the training distribution, as well as their element size, scale, or overall geometric variability, a prediction is proposed for a new artificial geometry with greater variability, as shown in Fig. 20a).

Fig. 18 Comparison of the final inference step between the model's prediction and the numerical solution, used as the ground truth, for untrained geometries with four imposed displacements. The color scale represents the 2-norm of the displacement vector at each node for the top comparison, while for the bottom comparison, it shows the value of the tangential stress, σ_{xy} (Pa). Both comparisons correspond to two separate test simulations, each with four imposed displacements, in a twelve-pass model trained with a maximum of three imposed actuators
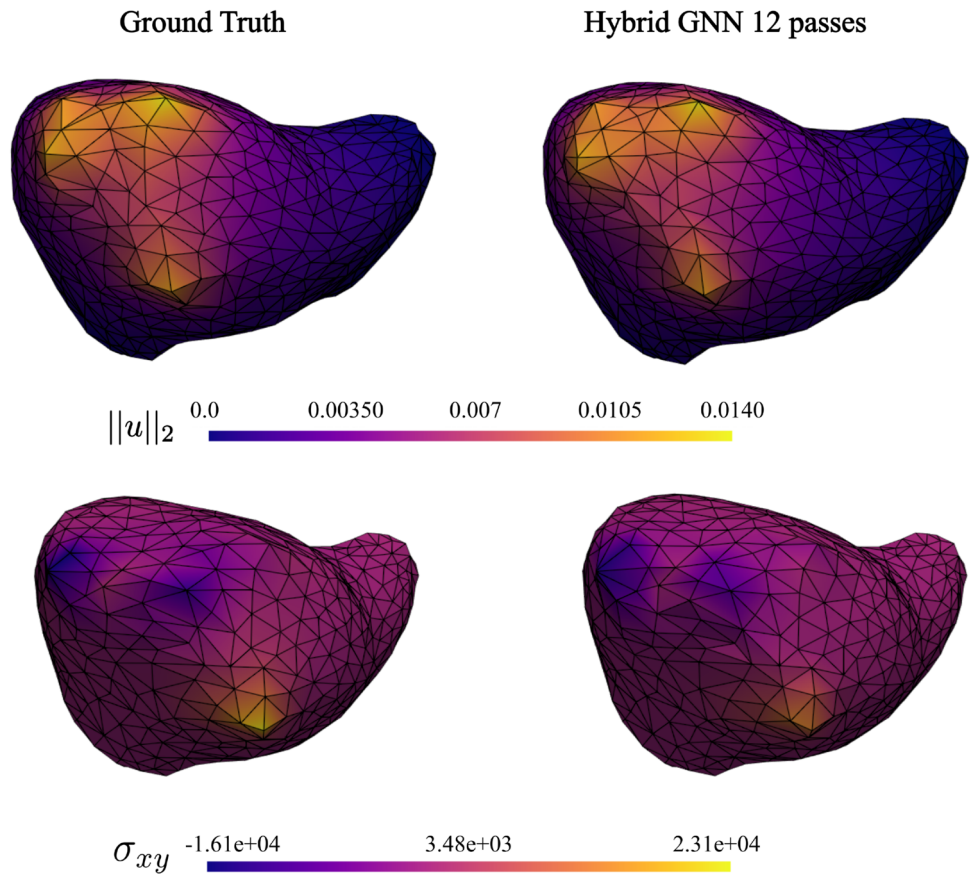
Fig. 19 Comparison of RRMSE rollouts between the test subset and the new extrapolation subset, with four imposed displacement points on the **L5** geometry
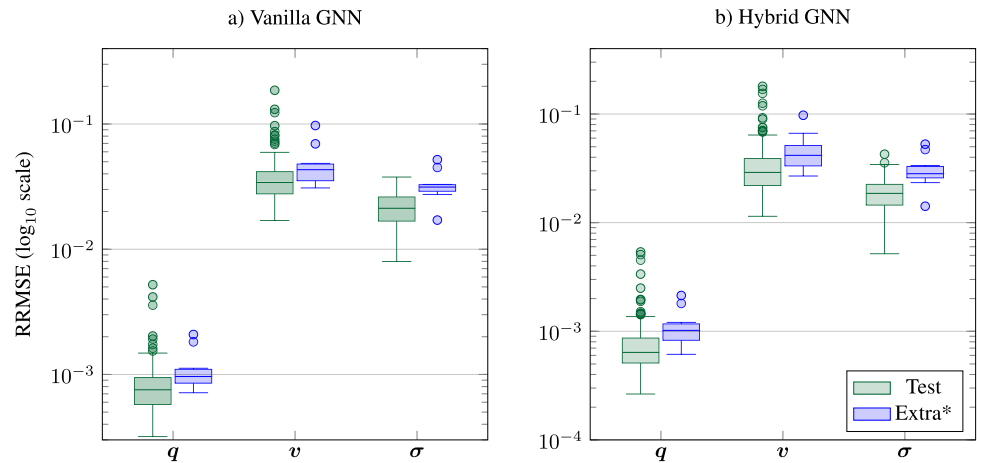
Fig. 20 Validation of the framework's extrapolation capabilities across strongly different anatomies to rigorously evaluate its generalization robustness and predictive fidelity
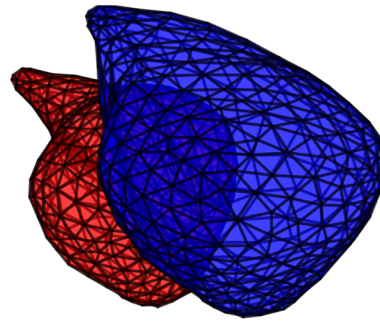
(a) Comparison between the **L1** mesh and an out-of-distribution, different geometry in both relative size—it is some 30% bigger in size—and shape.
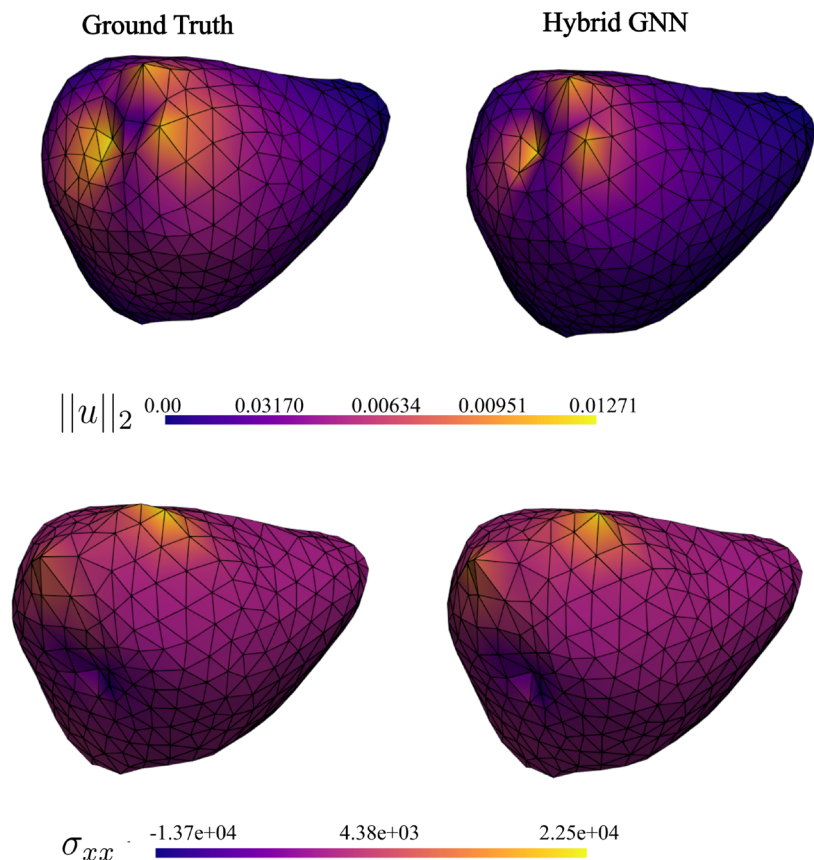
(b) Comparison of the final inference step between the proposed model and the numerical solution, used as ground truth, for the untrained artificial and scaled geometry

References

1. Ritto TG, Rochinha FA (2021) Digital twin, physics-based model, and machine learning applied to damage detection in structures. Mech Syst Signal Proc 155:107614
2. Lam R, Sanchez-Gonzalez A, Willson M, Wirnsberger P, Fortunato M, Alet F, Battaglia P (2022) Graphcast: Learning skillful medium-range global weather forecasting. arXiv preprint arXiv:2212.12794
3. Bruynseels K, Santoni de Sio F, van den Hoven J (2018) Digital twins in health care: ethical implications of an emerging engineering paradigm. Front Genet. https://doi.org/10.3389/fgene.2018.00031
4. Coorey G, Figtree GA, Fletcher DF, Snelson VJ, Vernon ST, Winlaw D, Redfern J (2022) The health digital twin to tackle cardiovascular disease-a review of an emerging interdisciplinary field. NPJ Digital Medicine 5(1):126
5. Corral-Acero J, Margara F, Marciniak M, Rodero C, Loncaric F, Feng Y, Gilbert A, Fernandes JF, Bukhari HA, Wajdan A et al (2020) The 'digital twin' to enable the vision of precision cardiology. Eur Heart J 41(48):4556–4564
6. Mendizabal A, Márquez-Neila P, Cotin S (2020) Simulation of hyperelastic materials in real-time using deep learning. Med Image Anal 59:101569
7. Stéphanie M, Simon C, Hervé D (2017) Nonlinear biomechanical model of the liver. In: Payan Y, Ohayon J (eds) Biomechanics of Living Organs. Elsevier, pp 243–265
8. Martínez Martínez F (2014) Determining the Biomechanical Behavior of the Liver Using Medical Image Analysis and Evolutionary Computation. PhD thesis, Universitat Politècnica de València
9. Tornero Costa R, Martinez Sanchís S (2020) Desarrollo de modelos biomecanicos del higado mediante metodo de elementos finitos
10. Zhou F, Molinari JF (2004) Dynamic crack propagation with cohesive elements: a methodology to address mesh dependency. Int J Numer Meth Eng 59(1):1–24
11. Ambati R, Yuan H (2011) Fem mesh-dependence in cutting process simulations. Int J Adv Manuf Technol 53:313–323
12. Li Z, Liu F, Yang W, Peng S, Zhou J (2021) A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 33(12):6999–7019
13. Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P (2017) Geometric deep learning: Going beyond euclidean data. IEEE Signal Process Mag 34(4):18–42
14. Medsker Larry R, Jain Lakhmi C (2001) Recurrent neural networks: design and applications, vol 5. CRC Press
15. Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW (2021) Learning mesh-based simulation with graph networks
16. Zonghan W, Pan S, Chen F, Long G, Zhang C, Yu Philip S (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32(1):4–24
17. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
18. Ece K (2016) Directions in hybrid intelligence: complementing ai systems with human intelligence. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)
19. Weinan E (2017) A proposal on machine learning via dynamical systems. Commun Math Stat 1(5):1–11
20. Greydanus S, Dzamba M, Yosinski J (2019) Hamiltonian neural networks. Adv Neural Inf Proc Syst
21. Galimberti CL, Furieri L, Liang X, Ferrari-Trecate G (2023) Hamiltonian deep neural networks guaranteeing nonvanishing gradients by design. IEEE Trans Autom Control 68(5):3155–3162
22. Bhattoo R, Ranu S, Anoop Krishnan NM (2023) Learning the dynamics of particle-based systems with lagrangian graph neural networks. Mach Learn: Sci Technol 4(1):015003
23. Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D, Ho S (2020) Lagrangian neural networks. arXiv preprint arXiv:2003.04630
24. Quercus H, Alberto B, Francisco C, Elias C (2022) Thermodynamics-informed graph neural networks. IEEE Trans Art Intel 5(3):967–76
25. Hernández Q, Badías A, González D, Chinesta F, Cueto E (2020) Structure-preserving neural networks. J Comput Phys 420:109950
26. Hernández Q, Badías A, Chinesta F, Cueto E (2023) Thermodynamics-informed neural networks for physically realistic mixed reality. Comput Methods Appl Mech Eng 407:115912
27. Pegolotti L, Pfaller MR, Rubio NL, Ding K, Brufau RB, Darve E, Marsden AL (2024) Learning reduced-order models for cardiovascular simulations with graph neural networks. Comput Biol Med 168:107676
28. Tierz A, Alfaro I, González D, Chinesta F, Cueto E (2024) Graph neural networks informed locally by thermodynamics. arXiv preprint arXiv:2405.13093
29. Hernández Q, Badías A, Chinesta F, Cueto E (2023) Port-metriplectic neural networks: thermodynamics-informed machine learning of complex physical systems. Comput Mech 72(3):553–561
30. Tierz A, Iparraguirre MM, Alfaro I, Gonzalez D, Chinesta F, Cueto E (2024) On the feasibility of foundational models for the simulation of physical phenomena. arXiv preprint arXiv:2410.14645
31. Cai S, Mao Z, Wang Z, Yang M, Karniadakis GE (2021) Physics-informed neural networks (PINNs) for fluid mechanics: a review. Acta Mech Sin 37(12):1727–1738
32. Faroughi S, Pawar NM, Fernandes C, Raissi M, Das S, Kalantari NK, Mahjour SK (2024) Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics. J Comput Inf Sci Eng 24(4):040802
33. Veličković P (2022) Message passing all the way up
34. Öttinger HC, Grmela M (1997) Dynamics and thermodynamics of complex fluids. II. illustrations of a general formalism. Phys Rev E 56(6):6633
35. Grmela M, Öttinger HC, Grmela and (1997) Dynamics and thermodynamics of complex fluids. Phys Rev E 56(6):6620
36. Guha P (2007) Metriplectic structure, leibniz dynamics and dissipative systems. J Math Anal Appl 326(1):121–136
37. Morrison PJ (1986) A paradigm for joined Hamiltonian and dissipative systems. Physica D 18(1–3):410–419
38. Öttinger HC (2006) Nonequilibrium thermodynamics for open systems. Phys Rev E 73:036126
39. J. Li et al (2023) MedShapeNet - A Large-Scale Dataset of 3D Medical Shapes for Computer Vision. arXiv:2308.16139
40. Ogden RW (1997) Non-linear Elastic Deformations. Courier Corporation
41. Park SW, Schapery R (1999) Methods of interconversion between linear viscoelastic material functions. part i-a numerical method based on prony series. Int J Solids Struct 36(11):1653–1675
42. Mena A, Bel D, Alfaro I, Gonzalez D, Cueto E, Chinesta F (2015) Towards a pancreatic surgery simulator based on model order reduction. Adv Model Simul Eng Sci 2:1–6

43. Quesada C, Alfaro I, Gonzalez D, Chinesta F, Cueto E (2018) Haptic simulation of tissue tearing during surgery. Int J Num Methods Biomed Eng 34(3):e2926

44. Urdeitx P, Alfaro I, Gonzalez D, Chinesta F, Cueto E (2024) A comparison of single and double generator formalisms for thermodynamics-informed neural networks. Comput Mech. https://doi.org/10.1007/s00466-024-02564-3

45. Kingma Diederik P, Welling M (2022) Auto-encoding variational bayes

46. Xu N, Wang P, Chen L, Tao J, Zhao J (2019) MR-GNN: multi-resolution and dual graph neural network for predicting structured entity interactions. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19

47. Son Hy T, Risi K (2022) Multiresolution equivariant graph variational autoencoder

48. Liu J, Hooi B, Kawaguchi K, Xiao X (2022) Mgnni: Multiscale graph neural networks with implicit layers

49. Moya B, Badías A, González D, Chinesta F, Cueto E (2023) A thermodynamics-informed active learning approach to perception and reasoning about fluids. Comput Mech 72(3):577–591

50. Bronstein Michael M, Bruna J, Cohen T, Veličković P (2021) Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, chapter GeometricDomains: the 5Gs