Virtual Reality (2025) 29:114
https://doi.org/10.1007/s10055-025-01199-w

ORIGINAL ARTICLE

A neural network architecture for physically-consistent haptic rendering

Quercus Hernández[1] · Pedro Martins[2] · Lucas Tesan[2] · Icíar Alfaro[2] · David González[2] · Francisco Chinesta[3,4] · Elías Cueto[2]

Received: 21 March 2024 / Accepted: 7 July 2025

Abstract

We develop a method for realistic haptic rendering of generalized solids that employs graph neural networks. In order to give these neural networks the required realism, biases of different types are used. In the case of solids with (hyper) elastic, hence reversible, behaviour, we use Hamiltonian neural networks. These networks employ a cognitive bias that ensures energy conservation during the simulation. In the case of dissipative solids (particularly visco-hyperelastic solids) we extend the cognitive bias so that it takes into account entropy production. In this way, regardless of the type of solid considered, the developed neural networks are able to provide the haptic devices with the necessary information for a realistic and, above all, physically consistent rendering. The results are tested using a set of haptic gloves and virtual reality glasses, which have shown excellent realism, but not without certain limitations due to the state of the art in the development of hardware of this type.

Keywords Real-time simulation · Haptic synthesis · Thermodynamics · Scientific machine learning

✉ Elías Cueto
ecueto@unizar.es

Quercus Hernández
quercus@seas.upenn.edu

Pedro Martins
pmartins@unizar.es

Lucas Tesan
ltesan@unizar.es

Icíar Alfaro
iciar@unizar.es

David González
gonzal@unizar.es

Francisco Chinesta
Francisco.Chinesta@ensam.eu

[1] Mechanical Engineering, University of Pennsylvania, Philadelphia 28006, USA

[2] ESI Group-UZ Chair of the National Strategy on Artificial Intelligence, Aragon Institute of Engineering Research, Universidad de Zaragoza, Maria de Luna, s.n., 50018 Zaragoza, Spain

[3] ESI Group Chair, PIMM Lab, ENSAM Institute of Technology, 151 Bvd. de l'Hôpital, 75013 Paris, France

[4] CNRS@CREATE LTD., ENSAM Institute of Technology, 1 CREATE Way, Singapore 138602, Singapore

1 Introduction

By haptic rendering we refer to a set of algorithms that compute an approximation to the physical interaction forces between the haptic interface representation inside a virtual environment and the virtual objects in that environment (Salisbury et al. 2004; Laycock and Day 2003). In general, haptic systems consist of a simulation engine to compute the time evolution of the virtual environment, a set of haptic (often accompanied by visual) rendering algorithms to compute the response of the environment to its simulated state, and finally a set of transducers, devices that convert the synthetic signals coming from the rendering algorithms into a form suitable for human perception (Otaduy et al. 2016; Laycock and Day 2007; Otaduy and Lin 2022).

While visual rendering enjoys a long scientific tradition that has led it to become a mainstream product today (think of the wide range of virtual or augmented reality glasses available on the market), haptic rendering is generally accepted as much more difficult to achieve. This is because, while continuous visual rendering requires synthesising images at a rate of about 25–60 Hertz, the haptic equivalent requires synthesising forces at a rate of 500-1000 Hertz. If you want to give the system a certain degree of physical realism, this

Published online: 22 July 2025

is equivalent to solving models, usually very complex and strongly non-linear, at least 500 times per second. This prevents the most commonly used techniques in solid mechanics, such as the finite element method (Belytschko 2007) or the boundary element method (Aliabadi 2020), from being used, at least in a standard way, despite the tremendous advances in terms of computing power.

The mechanical design of consumer products has evolved to such an extent that the first car designed entirely virtually is already on the road. In this case, crash tests were conducted on the first models to roll off the assembly line, such is the industry's confidence in today's simulation methods [8].

While simulation for mechanical design problems has been used for decades to achieve entirely virtual processes, as mentioned above, there are certain problems that have not yet achieved this degree of sophistication. If we refer to the world of the automotive industry, the perceived feel, for example, of the dashboard often acts as an indicator of product quality. The same problem applies to the feel of the body sheet: a sheet that buckles under slight pressure will be rejected by consumers, but this behaviour depends more on the geometry of the sheet than on its thickness.

Another industry that relies heavily on the sensation perceived by the sense of touch is cosmetics. Industrialists in this sector report that the viscosity, degree of elasticity, etc. of cosmetic creams play a fundamental role in consumer choice. Solving this problem experimentally requires costly combinations of chemicals to achieve different perceived characteristics of the products, which is a headache for the industry in this sector.

A completely different sector in which the sensation of touch plays a key role is surgery (Maciel et al. 2009; Courtecuisse et al. 2010). Surgeons using endoscopic techniques must learn to see the organs they are working on through a monitor, thus losing the depth vision that does not exist in a flat image. In this context, the sense of touch, perceived through the contact of the surgical instruments with the patient's organs, plays a fundamental role in their practice, as well as in their training. This is why endoscopic surgery is one of the sectors where there has been the greatest interest in the development of physically realistic haptic rendering algorithms.

With the emergence of (deep) neural networks, and particularly physics-informed neural networks (PINNs), new possibilities for efficient problem solving are opening up (Raissi et al. 2019; Joolee and Jeon 2021). While PINNs have been a tremendous success in our community, they have certain limitations for the application at hand. For example, they rely on training the network based on the fulfilment of a certain differential equation, which must therefore be known in advance. This is not always the case. Applications abound in which only data on the behaviour of the physical system are available, but the details of the differential equation governing the phenomenon in question are unknown.

In this paper we explore the development of techniques that allow the use of synthetic data from high-fidelity simulation, in our case using finite elements. Since these data, due to their origin, have a mesh structure, this fact will be used to establish a parallelism with a graph, exploiting the advantages that graph networks have shown to have (Pfaff et al. 2020).

However, to avoid as much as possible the black box character of networks trained on data alone, in this paper we will explore the advantages of using inductive biases to ensure that certain basic principles are met. As will be seen below, a formalism is developed that allows us to unify the treatment of both conservative and dissipative materials, ensuring correct compliance with the laws of thermodynamics (conservation of energy and entropy production). This will ensure the realism of both the deformation field obtained for the visual rendering of the phenomenon and the prediction of the contact force to be transmitted to the haptic device.

The paper is organized as follows. In Sect. 2 we overview existing approaches in the field, with emphasis on physically-consistent methods. Our proposed method is described in Sect. 3, where we describe how we use Graph Neural networks for the objectives of this work. In Sect. 4 we deeply analyse the performance of the proposed technique on three different problems. Namely, a purely (nonlinear) elastic problem of shell buckling and post-buckling, a visco-hyperelastic material's haptic response and, finally, an application to surgery, where liver soft tissue is considered. The paper ends with an analysis of these results in Sect. 5.

2 Related work

Haptic rendering, although a considerably more complex problem than visual rendering in many respects, has a long tradition in the scientific literature. There are a number of high quality reviews that the reader can consult at Laycock and Day (2003), Otaduy et al. (2016), Otaduy and Lin (2022), Laycock and Day (2007), Salisbury et al. (2004). As mentioned above, a particularly active field of haptic rendering research is focused on the development of endoscopic surgery simulators (Deo and De 2009; Maciel et al. 2009; Rangarajan et al. 2020; Escobar-Castillejos et al. 2016; Panait et al. 2009).

In this case, in addition to the difficulties inherent to haptic rendering, already mentioned, the complexity of the physical phenomena to be reproduced must be added:

non-linearity of living tissues, multiple contacts between organs, cutting and tearing phenomena, etc. These difficulties soon prompted the exploration of techniques that did not involve the complete resolution of the complex systems of non-linear equations to which the finite element method gives rise (although there are approximations to the problem that employ, so to speak, brute force, see Miller et al. (2007), for instance).

Such alternative techniques to finite elements include those constructed using model reduction methods (Barbič and James 2007; Niroomandi et al. 2008, 2013a, b; Quesada et al. 2017). This included the simulation of surgical cutting, for instance, Niroomandi et al. (2011), Quesada et al. (2016). Other works in this same research line include (González et al. 2015). A recent review of model order techniques for surgery simulation can be found at Quesada et al. (2023).

In a parallel vein, the use of neural networks for haptic rendering also has a long history (Joolee and Jeon 2021; Heravi et al. 2024; Vlachos and Moustakas 2024). Initially, neural networks have been used for understanding the haptic perception process, not so much for rendering, with rigid objects (Johnsson and Balkenius 2007). With the popularization of deep learning, the applications have become numerous. For example, in Joolee and Jeon (2021) the authors employ neural networks trained from experimentally obtained data to obtain realistic haptic responses. In Chen et al. (2016) a method is presented for the interaction with deformable media whose nonlinearity is smooth and which, nevertheless, presents quite appreciable deviations with respect to the results obtained by simulation. Strictly related to this, in Lang and Andrews (2010) is presented a method for the experimental acquisition of data on roughness of different surfaces that could well be used for the training of neural networks for this purpose.

In general, despite what has been said above, there is not a wide tradition about the use of neural networks in haptic rendering. This is evidenced by some recent reviews, such as Otaduy and Lin (2022), for example.

In what follows, we will show the efforts devoted to create different inductive biases to ensure the physical validity of what is rendered, both visually and haptically.

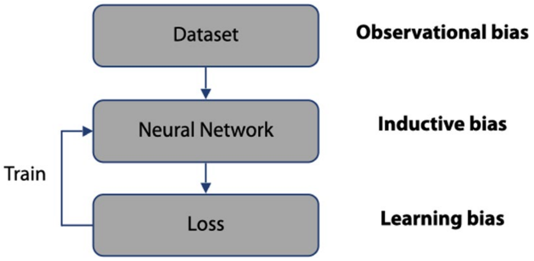


Fig. 1 Different types of biases in the architecture of a neural network

3 Neural network architecture

3.1 Thermodynamic inductive biases

There is a nice hypothesis that human intelligence is based on a few inductive biases rather than a long list of heuristics (Goyal and Bengio 2022). In general, in the field of deep learning, different types of bias are admitted, as shown in Fig. 1. Observational bias refers to the set of techniques that use a particular data collection scheme to enhance learning. This type of bias includes, for example, data augmentation techniques.

An inductive bias is the set of assumptions that a machine learning algorithm makes about the relationship between input data and output variables. Among these, one can cite, for instance, convolution, if one wants to enforce group equivariance, or recurrence for equivariance over time. These biases help the learning process to reach, as far as possible, the global minimum of the loss function and not one of the multiple local minima that are often possible.

Finally, learning bias refers to techniques that modify the loss function to take into account some type of information to be retained during the inference phase. The ubiquitous example of this type of bias is that employed by PINNs (Raissi et al. 2019), which impose the fulfillment of the differential equation governing the physical phenomenon under study by adding the remainder of it to the loss function.

From our humble point of view, the most interesting biases are the inductive ones. In general, observational biases involve an inordinate increase in the size of the data set and, in turn, learning biases often involve networks whose expressivity is constrained and which are therefore difficult to train. If it is possible to codify the general principles that the network must satisfy and introduce them into the network architecture itself, the result is generally a higher accuracy in the predictions obtained and a smaller size of the data set needed for training.

In our approach, we consider a vector $\boldsymbol{z} \in \mathcal{M} \subseteq \mathbb{R}^n$ of state variables governing the physics of the problem. $\mathcal{M}$ represents the set (which is assumed to have a manifold structure) of all the admissible states of the system. The simplest possible evolution of the system will take a form

$$\dot{\boldsymbol{z}} = \frac{d\boldsymbol{z}}{dt} = F(\boldsymbol{z}, t), \ t \in \mathcal{I} = (0, T], \ \boldsymbol{z}(0) = \boldsymbol{z}_0, \tag{1}$$

where t refers to the time instant within an interval $\mathcal{I}$ and $F(\boldsymbol{z}, t)$ refers to an arbitrary nonlinear differential function, which is to be found from data.

The type of inductive bias that we employ consists precisely in assuming a certain form for the function $F(\boldsymbol{z}, t)$. For example, if the phenomenon whose haptic response is to be synthesised is known to be conservative (hence reversible, as in the case of elastic materials), it can be assumed that the evolution of the variables that govern it will have a symplectic form, i.e,

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}}, \tag{2}$$

where matrix $\boldsymbol{L} = \boldsymbol{L}(\boldsymbol{z})$ represents the so-called Poisson matrix and is required to be skew-symmetric. E represents the energy of the (closed) system. This idea has given rise to multiple implementations of what are known as Hamiltonian neural networks, Greydanus et al. (2019). It would therefore be sufficient to learn from data the particular form of $\boldsymbol{L}$ and the potential E to have fully characterised the evolution of the system. This approximation would be sufficient to predict the behaviour of elements consisting of elastic sheets, provided that they do not enter the plastic regime, for example.

However, many materials of industrial interest, as well as almost all soft biological tissues, exhibit certain levels of dissipation. In this case, it will be necessary to assume that the free energy of the system takes the form $\mathcal{F} = E + S$, with S a generalized entropy potential. In that case,

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}}, \tag{3}$$

holds if we require these potentials to satisfy the so-called degeneracy conditions,

$$\boldsymbol{L}\frac{\partial S}{\partial \boldsymbol{z}} = \boldsymbol{M}\frac{\partial E}{\partial \boldsymbol{z}} = \boldsymbol{0}. \tag{4}$$

Here, $\boldsymbol{M}$ is the so-called the friction matrix, a matrix describing the system's irreversible behaviour. In other words, energy has nothing to do with dissipation phenomena and, in turn, entropy is not related to the conservation of energy. Eqs. (3) and (4) constitute a so-called metriplectic formalism (since it is, at the same time, symplectic in energy and follows a metric flow for entropy) often referred to as GENERIC (an acronym for General Equation for the Non-Equilibrium Reversible-Irreversible Coupling, Grmela and Öttinger (1997), Öttinger and Grmela (1997). It is not difficult to show that if the GENERIC formalism is satisfied, conservation of energy in closed systems and non-negative entropy production is ensured.

In what follows we assume, without loss of generality, that data on the behaviour of the system with a periodicity Δt has been sampled, so that the equation can be approximated by

$$\frac{d\boldsymbol{z}}{dt} \simeq \frac{\boldsymbol{z}_{t+1} - \boldsymbol{z}_t}{\Delta t} \quad \longrightarrow \quad \boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \Delta t\left(\boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}_t} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}_t}\right). \tag{5}$$

The skew-symmetric and positive semi-definite conditions applying on $\boldsymbol{L}$ and $\boldsymbol{M}$, respectively, are enforced by learning instead two matrices $\boldsymbol{l}$ and $\boldsymbol{m}$, so that

$$\boldsymbol{L} = \boldsymbol{l} - \boldsymbol{l}^{\top}, \qquad \boldsymbol{M} = \boldsymbol{m}\boldsymbol{m}^{\top}. \tag{6}$$

The fulfillment of the degeneracy conditions will be enforced through a learning bias, as will be detailed below.

Eqs. (3) and (4) constitute a completely general formalism, valid for any type of physical or material phenomenon considered, with the exception that the dissipation potential is assumed to be quadratic. For the case of a material in the plastic range, for example, a slightly more general form of the dissipative term must be used, although it is well known in the literature. If it is known a priori that the phenomenon to be described is conservative or reversible, then the form given by Eq. (2) should be preferred because of its simplicity.

A final detail that should be mentioned is the possibility that the system is being actuated from the outside, thus ceasing to be a closed system. In that case, Eq. (3) must be modified to take this fact into account, giving rise to a formalism which is closely reminiscent of the port-Hamiltonian schemes, but which includes dissipation:

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}} + \boldsymbol{M}\frac{\partial S}{\partial \boldsymbol{z}} + \boldsymbol{F}, \tag{7}$$

where $\boldsymbol{F}$ represents a vector with the value of the externally imposed variables (Desai et al. 2021).

One of the main advantages of the GENERIC formalism is that it allows a lot of freedom in the choice of state variables. The vector $\boldsymbol{z}$ must collect the variables that allow to define the energy of the system. In principle, any set of variables is valid, even if it includes redundant variables, provided that it is capable of describing the energy of the system. This characteristic gives great flexibility to the mathematical description of the evolution of the system.

3.2 Geometric biases

The other major ingredient of our implementation exploits the structure of synthetic data, obtained through high-fidelity finite element simulations. In this way, the data already have a certain geometric structure, which derives from the mesh used to obtain them. It would seem natural, therefore, to use a graph neural network to take full advantage of this structure (Sanchez-Gonzalez et al. 2020; Tesán et al. 2024). The implementation of our graph networks closely follows the MeshGraphNet developed in Pfaff et al. (2020)

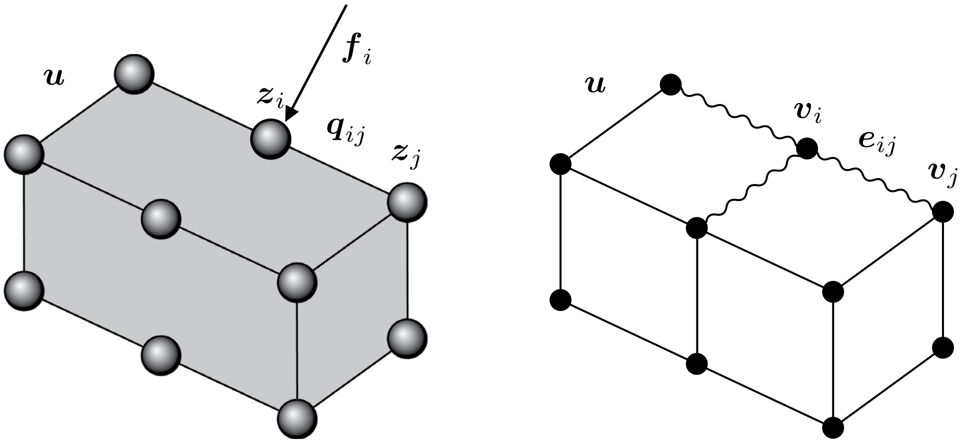


Fig. 2 Parellelism between a finite element mesh (left) with node state variables $\boldsymbol{z}_i$, relative nodal distances $\boldsymbol{q}_{ij}$, external interactions $\boldsymbol{f}_i$ and global properties $\boldsymbol{u}$ and its equivalent graph representation, with node and edge attributes: $\boldsymbol{v}_i$ and $\boldsymbol{e}_{ij}$(right)

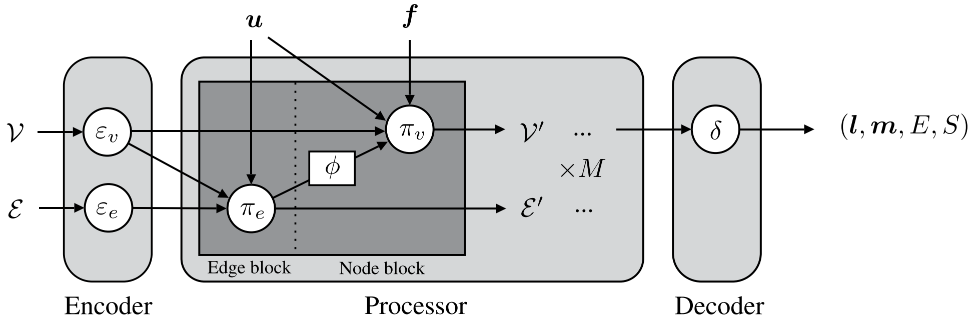


Fig. 3 Encode-process-decode scheme used in the algorithm. The encoder transform the node and edge features to a learnt embedding. The processor shares the nodal information through the graph via M message passing modules. The encoder extracts the GENERIC operators and potentials from the processed node embeddings

but has been suitably modified to take full advantage of the thermodynamic inductive biases discussed in the preceding section. In addition, the use of graph networks makes it possible to impose invariance and equivariance on those variables whose behaviour is known to follow these rules (Keriven and Peyré 2019).

We assume in our data a geometric structure given by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{u})$, with $\mathcal{V} = \{1, \dots, n\}$ is a set of $|\mathcal{V}| = n$ vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of $|\mathcal{E}| = e$ edges and $\boldsymbol{u} \in \mathbb{R}^{F_g}$ is a vector containing global features of the problem at hand. In our implementation we establish a parallelism between the vertices of the graph and the nodes of the finite element mesh, with edges playing a similar role to nodal connectivity. The vector of global characteristics incorporates parameters that globally affect the whole model, such as the value of gravity, Young's modulus, etc. For each vertex $i \in \mathcal{V}$ we associate a feature vector $\boldsymbol{v}_i \in \mathbb{R}^{F_v}$, which stores physical properties of each individual node. In the same spirit, for each edge $(i, j) \in \mathcal{E}$ we associate an edge feature vector $\boldsymbol{e}_{ij} \in \mathbb{R}^{F_e}$.

In order to ensure Galilean invariance of the results, we associate to the edges $\boldsymbol{e}_{ij}$ the positional state variables of the system ($\boldsymbol{q}_i$) but, notably, the relative distance among nodes, $\boldsymbol{q}_{ij} = \boldsymbol{q}_i - \boldsymbol{q}_j$. The rest of the state variables are assigned to the node feature vector $\boldsymbol{v}_i$. The external interactions, such as forces applied to the system, are included in an external load vector $\boldsymbol{f}_i$ see Fig. 2.

Our implementation follows the already classic encode-process-decode architecture in Pfaff et al. (2020). In essence, the graph networks share one MLP for the vertices and another MLP for the edges. Its architecture is sketched in Fig. 3. It consists of the following ingredients:

- Encoder: two MLPs are employed, ε_v and ε_e, to transform the vertex and edge initial feature vectors into higher-dimensional embeddings $\boldsymbol{x}_i \in \mathbb{R}^{F_h}$ and $\boldsymbol{x}_{ij} \in \mathbb{R}^{F_h}$ respectively.

$$\begin{aligned} \varepsilon_e : \mathbb{R}^{F_e} &\longrightarrow \mathbb{R}^{F_h} & \varepsilon_v : \mathbb{R}^{F_v} &\longrightarrow \mathbb{R}^{F_h} \\ \boldsymbol{e}_{ij} &\longmapsto \boldsymbol{x}_{ij} & \boldsymbol{v}_i &\longmapsto \boldsymbol{x}_i \end{aligned}$$

- Processor: The processor is responsible for sharing the nodal information between vertices via message passing. It modifies the hidden vectors so as to extract the sought output of the system. An MLP (π_e) computes the updated edge features $\boldsymbol{x}'_{ij}$ for each graph edge, based on the current edge features, global features, and sending and receiving node. In a second step at each node the messages are pooled with a permutation invariant function ϕ such as sum or mean, based on the neighborhood $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ of the node i. Finally, the node embeddings are updated with a second MLP (π_v) using the current node features, the pooled messages, the external load vector and the global features:

$$\begin{aligned} \pi_e : \mathbb{R}^{3F_h+F_g} &\longrightarrow \mathbb{R}^{F_h} & \pi_v : \mathbb{R}^{2F_h+F_f+F_g} &\longrightarrow \mathbb{R}^{F_h} \\ (\boldsymbol{x}_{ij}, \boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{u}) &\longmapsto \boldsymbol{x}'_{ij} & (\boldsymbol{x}_i, \phi(\boldsymbol{x}'_{ij}), \boldsymbol{f}_i, \boldsymbol{u}) &\longmapsto \boldsymbol{x}'_i \end{aligned}$$

where $(\cdot, \cdot)$ denotes vector concatenation and $\boldsymbol{x}'_i$ and $\boldsymbol{x}'_{ij}$ are the updated nodal and edge latent vectors. To transmit the information along the network geometry, the message passing process needs to be repeated with both shared or unshared parameters in M processing blocks.

- Decoder: The procedure ends by extracting the sought physical output $\boldsymbol{y}_i \in \mathbb{R}^{F_y}$ of the system from the nodal latent feature vector, through an MLP (δ_v).

$$\begin{aligned} \delta_v : \mathbb{R}^{F_h} &\longrightarrow \mathbb{R}^{F_y} \\ \boldsymbol{x}'_i &\longmapsto \boldsymbol{y}_i = (\boldsymbol{l}, \boldsymbol{m}, E, S) \end{aligned}$$

3.3 Learning procedure

Synthetic data coming from high-fidelity finite element simulation $\mathcal{D}$ are composed by N_{sim} different loading cases of a system that evolves in time. Some of the studied examples are considered quasi-static—although evolving in time—, while others incorporate the inertia typical of dynamical simulations. Our method is trained through data pairs $\mathcal{D}_i$ containing $\boldsymbol{z}_t$ and its evolution in time $\boldsymbol{z}_{t+1}$ for each node of the finite element mesh/graph,

$$\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^{N_{\text{sim}}}, \quad \mathcal{D}_i = \{(\boldsymbol{z}_t, \boldsymbol{z}_{t+1})\}_{t=0}^{T}. \tag{8}$$

We partition $\mathcal{D}$ in 80% training, 10% test and 10% validation sets: $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$ respectively.

Training is performed such that our method operates as a time-integration scheme. The loss function is composed by:

- Data loss: that takes into account the accuracy in the prediction of the training data. We employ the MSE between the predicted and the ground-truth time evolution of the state variables,,

$$\mathcal{L}_n^{\text{data}} = \left\| \frac{d\boldsymbol{z}^{\text{GT}}}{dt} - \frac{d\boldsymbol{z}^{\text{net}}}{dt} \right\|_2^2, \tag{9}$$

where $\| \cdot \|_2$ denotes the L2-norm. In some of our previous work, the loss term included only the error norm in the $\boldsymbol{z}$ variable itself and not in its derivative. In this paper we have decided to use the time derivative of $\boldsymbol{z}$, as this is convenient, given the time integrator form of our method, see Eq. (5). Indeed, a form of Sobolev-type training could be included, by including the second derivative of $\boldsymbol{z}$, but our experience indicates that there is no appreciable improvement.

- Degeneracy loss: For dissipative systems (this is not necessary in conservative examples), we add a second term to enforce the fulfillment of the degeneracy conditions. This second term is the MSE of the residual terms corresponding to the energy and entropy degeneracy conditions,

$$\mathcal{L}_n^{\text{deg}} = \left\| \boldsymbol{L} \frac{\partial S}{\partial \boldsymbol{z}_n} \right\|_2^2 + \left\| \boldsymbol{M} \frac{\partial E}{\partial \boldsymbol{z}_n} \right\|_2^2. \tag{10}$$

The global loss term is finally obtained by weighting the two terms over the N_{batch} batched snapshots.

$$\mathcal{L} = \frac{1}{N_{\text{batch}}} \sum_{n=0}^{N_{\text{batch}}} (\lambda \mathcal{L}_n^{\text{data}} + \mathcal{L}_n^{\text{deg}}), \tag{11}$$

with λ a weighting parameter that helps to balance the influence of both losses in the final result.

4 Numerical results

For the examples below, two types of haptic devices are considered. On the one hand, a device focused on the use of a pair of Senseglove Nova haptic gloves (Senseglove, Molengraaffsingel 12, 2629JD Delft, The Netherlands). The equipment is also composed by a Vive Pro 2 Headset, VIVE Link BOX (2.0), 2 Controllers (2018), 2 base stations 2.0 and an HTC Vive tracker 3.0. The equipment is completed with a personal computer featuring a 13th Gen Intel(R) Core(TM) i7-13700, 2100 MHz processor, with 16 main processor is employed. It uses a GPU NVIDIA GeForce RTX 3060, 16GB RAM and an SSD drive Kingston SNV2S1000G 512GB. This equipment is shown in Fig. 4.

The second setup consists of a GeoMagic Sensible haptic peripheral (Geomagic. OpenHaptics Toolkit. 3D systems. Geomagic solutions, 430 Davis Drive, Suite 300 Morrisville, NC 27560, USA). It is a 3-degree-of-freedom device widely employed for the simulation of endoscopic surgery, among many other possible applications, see Fig. 5. Details of this prototype can be found at Quesada et al. (2016, 2017, 2023). The main difference is that, while the GeoMagic haptic peripheral has six degrees of freedom (three for haptic feedback only), the SenseGlove gloves include four sensors to capture the flexion and extension of the thumb, the index, the middle and the ring fingers, plus one sensor to capture the abduction and adduction of the thumb. These movements are captured by measuring the extension of the cables on the glove. The perceived sensation of realism is therefore very different.

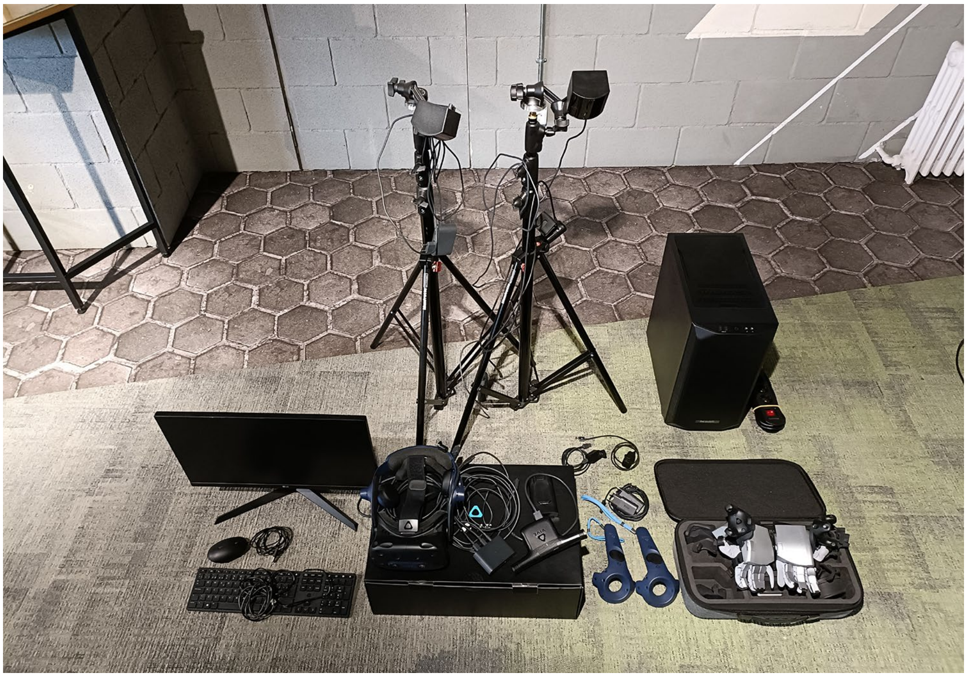


Fig. 4 Equipment for the Senseglove haptic gloves set

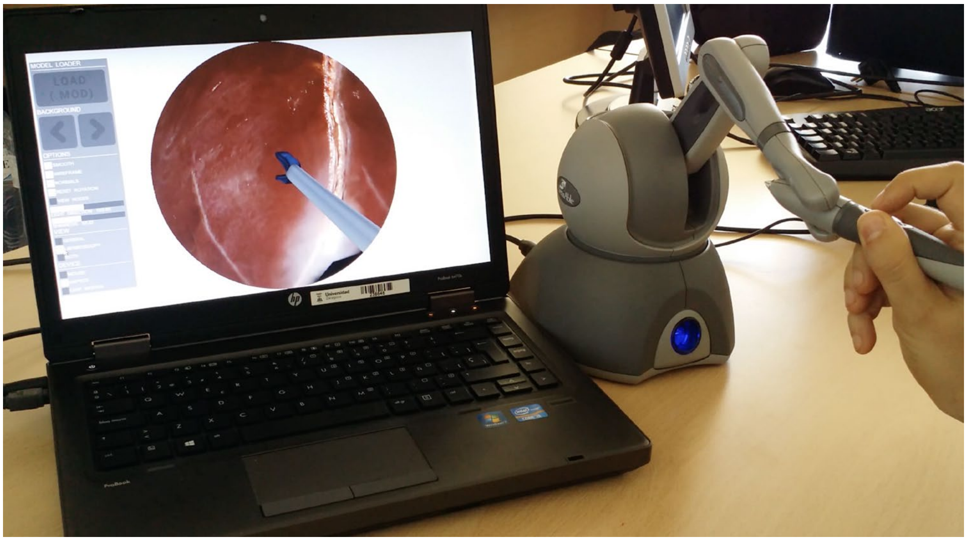


Fig. 5 The UniZar endoscopic surgery simulator, equipped with a Geomagic haptic device

4.1 Prediction of shell snap-through

The tactile sensation offered by metallic shells is of great importance in the design of many consumer products. Among them, certainly of no small importance, is the bodywork of automobiles, for example. To test the accuracy and robustness of our approximation, we begin by testing the behaviour of a sheet under the action of a point force at an arbitrary point on its surface. At all times it will be assumed that the shell does not enter a plastic regime, so that its behaviour will always be elastic, but taking into account the possibility of large displacements and deformations. We will study with particular attention the buckling and post-buckling phenomena.

Therefore, Eq. (7) will be reduced in this first example to

$$\frac{d\boldsymbol{z}}{dt} = \boldsymbol{L}\frac{\partial E}{\partial \boldsymbol{z}}, \tag{12}$$

so as to allow perfectly reversible effects.

The geometry of the sheet to be considered is shown in Fig. 6. Clamped boundary conditions are considered at the dashed edges. The shell is actually a shallow arc of 0.5×0.5 m^2 in area and 0.0125 m in height at the centre. It has a uniform thickness of 0.001 m. Synthetic data for this problem are obtained by finite element simulation. Material parameters for this example were Young's modulus $3.103 \cdot 10^9$ N/mm^2 and Poisson's ratio 0.3.

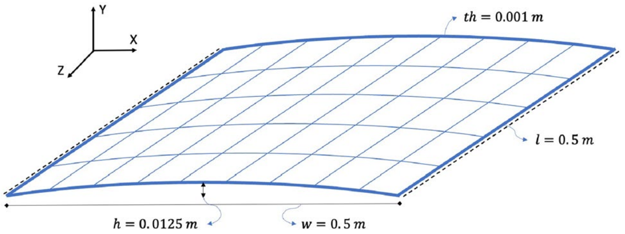


Fig. 6 Geometry of the shell snap-through problem. Dashed lines indicate clamped boundary conditions

The training database consisted of a total of 54 experiments with different load application points, out of a possible total of 864 possible positions. For them, a total of 1000 snap-through trajectory time instants were obtained. Of these 1000 time instants, only 100 were subsequently retained by pruning to keep the database to a manageable size.

It is expected that the developed method will be able to predict the displacement field of the shell for any load position not previously seen in the training. The load was considered to be ramped over a pseudo-time interval of 1 s. Finite element simulations employed pseudo-time increments of 0.001 s, that were subsequently pruned so as to extract one snapshot out of each ten possible (thus, each 0.06 s).

Figure 7 shows different time instants of the simulation, where the degree of accuracy achieved can be noticed.

Figure 8 shows the time evolution of the relative root mean squared error (RRMSE) over this interval for an arbitrary load position, not included in the training set. As can be seen, the error remains below 5% throughout the simulation.

An error of less than 5% in the displacement field is sufficient to provide the user with a realistic haptic feeling of the behaviour of the shells (Otaduy Tristán 2004).

4.2 Squashing of visco-hyperelastic foams

We consider the equations of equilibrium of a solid expressed in total Lagrangian form

$$\boldsymbol{\nabla P} + \boldsymbol{B} = \rho \ddot{\boldsymbol{u}} \text{ in } \Omega_0, \tag{13}$$

where following standard notation, $\boldsymbol{B}$ stands for the volumetric force applied to the body and $\boldsymbol{P}$ represents the first Piola-Kirchhoff stress tensor. $\Omega_0 = \Omega(t=0)$ is the undeformed configuration of the body. The problem is completed by enforcing appropriate boundary conditions,

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{X}) &= \overline{\boldsymbol{u}} \quad \text{on } \Gamma_u, \\ \boldsymbol{PN} &= \overline{\boldsymbol{t}} \quad \text{on } \Gamma_t, \end{aligned}$$

where Γ_u and Γ_t represent, respectively, the essential (Dirichlet) and natural (Neumann) portions of the boundary $\Gamma = \partial\Omega$ of the solid. We employ $\boldsymbol{X}$ to refer to the undeformed configuration, with $\boldsymbol{N}$ the unit vector normal to $\Gamma = \partial\Omega_0$. Finally, $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{t}}$ represent the applied displacement and traction, respectively.

To consider a solid with a sufficiently general behaviour for the purposes of this work, we assume that it is given by a combination of hyperelastic behaviour,

$$\boldsymbol{S} = \frac{\partial \Psi}{\partial \boldsymbol{E}}, \tag{14}$$

and a viscous component that assumes a variable shear relaxation modulus that is approximated via a one-term Prony series,

$$g_R(t) = 1 - \bar{g}_1(1 - e^{\frac{-t}{\tau_1}}),$$

with relaxation coefficient of $\bar{g}_1 = 0.3$ and relaxation time of $\tau_1 = 0.1$ s. $\boldsymbol{S}$ represents the second Piola-Kirchhoff stress tensor and $\boldsymbol{E}$ is the Green-Lagrange strain tensor.

The aim of the method developed in this paper is to solve the problem given by Eq. (13) under the very stringent real-time requirements of our haptic gloves, while maintaining the physical consistency of the results.

As a toy example, we consider a bunny mesh from the Stanford 3D scanning repository, see Fig. 9. The model from which synthetic data were obtained consisted of $N_e = 4941$ tetrahedral linear elements and $N = 1352$ nodes.

We assume that the strain-energy potential takes the neo-Hookean form

$$\Psi = C_{10}(I_1 - 3) + \frac{1}{D_1}(J_{e\ell} - 1)^2, \tag{15}$$

with $C_{10} = 2.6 \cdot 10^{-1}$ and $D_1 = 4.9 \cdot 10^{-2}$. I_1 represents the first invariant of the strain tensor, and $J_{e\ell}$ the volume change at a given point.

For the construction of the data set, a concentrated load of $F = 1N$ is applied in 100 different positions of the bunny, normal to the solid surface. The bunny was clamped at the ground plane by disabling displacements and rotations at the lower model nodes. Each simulation is composed of 20 time increments of $\Delta t = 5 \cdot 10^{-2}$ s.

The code is implemented in Python using the PyOpenGL wrapper for the visualization and Pytorch Geometric (Fey and Lenssen 2019) for the deep learning training and evaluation. Errors obtained with respect to a high fidelity finite element simulation on the same mesh are shown in Fig. 10. The appearance of the developed system is shown in Fig. 11.

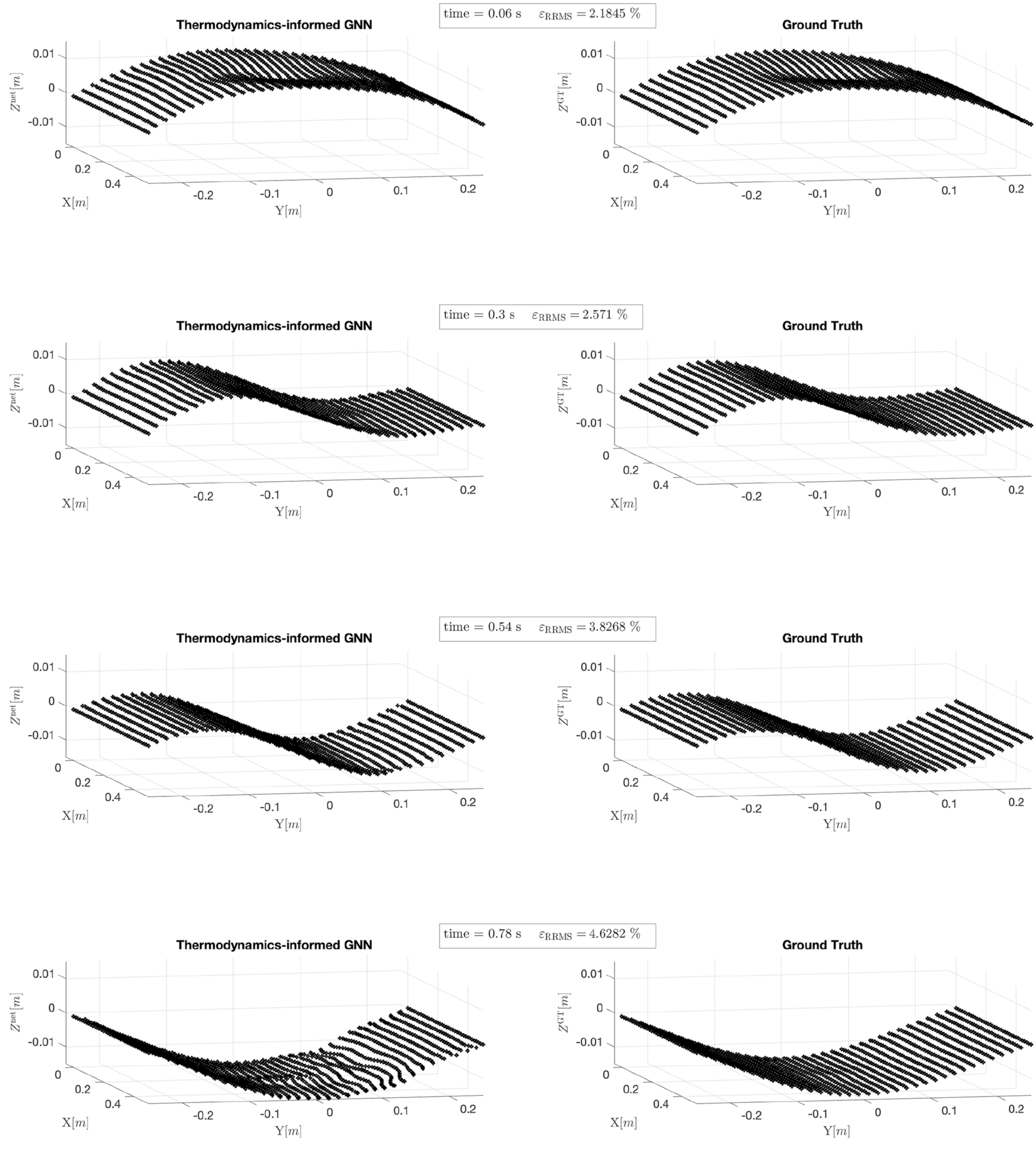


Fig. 7 Snapshots at different time instants across the shell snap-through simulation

4.3 Endoscopic liver surgery

We consider a simulation of cholecystectomy, i.e. surgery to remove the gallbladder. We focus in this case on the effects of palpation and manipulation of the liver, which is always necessary, given the location of the gallbladder below the liver.

The model of the liver is composed by 530 nodes, and 1734 linear tetrahedral finite elements. Its anatomy can be found in Fig. 12. For the constitutive model of liver parenchyma, an Ogden model with $N = 1$ term was employed,

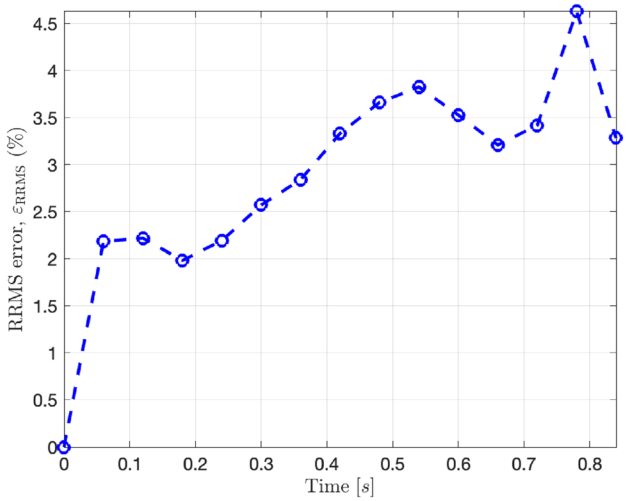


Fig. 8 Error during the shell snap-through simulation

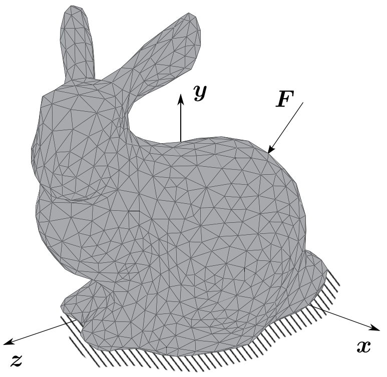


Fig. 9 Bunny mesh geometry. Nodes at its base are assumed to be clamped

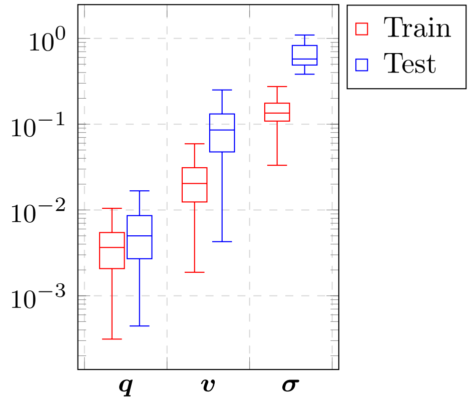


Fig. 10 Box plots for the relative error during a complete simulation of the bunny dataset in both train and test cases. The state variables employed for the description of the evolution of the energy in the system are position ($\boldsymbol{q}$), velocity ($\boldsymbol{v}$), and the stress tensor ($\boldsymbol{\sigma}$)

following Martínez (2014). In this model, the strain-energy potential takes the form

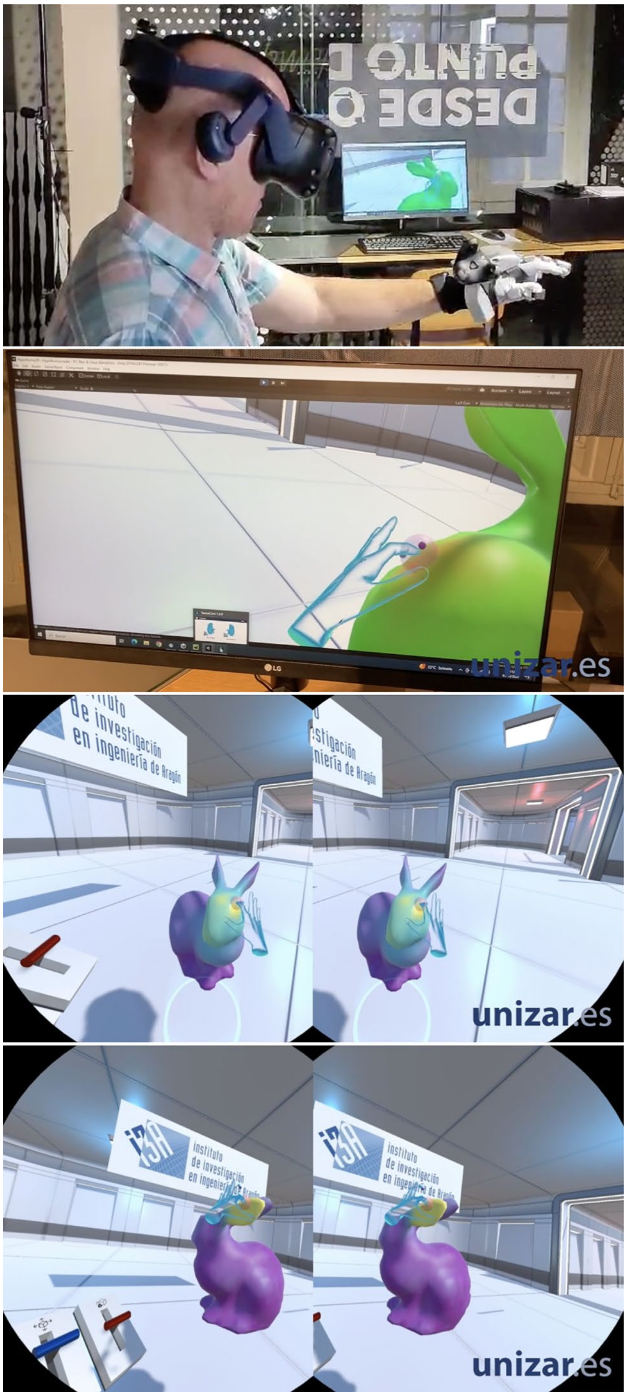


Fig. 11 Different views of the implemented setup. From top to bottom and left to right, appearance of the hardware (Vive Pro 2 headset and Senseglove Nova haptic gloves); displacement map as seen on the PC screen; two different instants of the stereographic views generated for the Vive Pro 2 headset, with a colourmap of the stress. The complete video can be seen from https://www.youtube.com/watch?v=0dETGlA0KeA

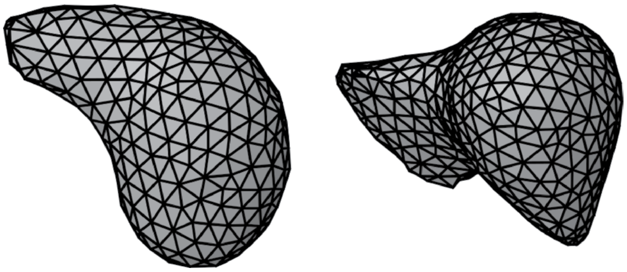


Fig. 12 Finite element model of the liver employed to obtain synthetic data

$$\Psi = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2} \left(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3\right) + \sum_{i=1}^{N} \frac{1}{D_i} (J_{e\ell} - 1)^{2i}, \qquad (16)$$

with $\mu_1 = 364.74$ Pa, $\alpha_1 = 16.19$, $D_1 = 9.525 \cdot 10^{-5}$ Pa^{-1} and $\bar{\lambda}_j$ the three principal stretches. The viscous character of the behaviour was introduced through a Prony series with two characteristic relaxation terms, $\bar{g}_1 = 0.56057$, $\bar{g}_2 = 0, 40585$, $\tau_1 = 6.36173$ s and $\tau_2 = 0.73348$ s.

The synthetic dataset was obtained through 85 simulations of loads applied to different points in the surface. These loads were assumed to be perpendicular to the surface of the liver at each location. 80% of them was employed, as it is usual, to train, and the subsequent 10% to validation, with the remaining 10% for testing purposes.

At a given snapshot of a palpation simulation, the difference between the ground truth stress field and the learned simulation is barely visible to the human eye, see Fig. 13. A quantitative evaluation of the accuracy of these results is plotted in Fig. 14. A boxplot of these errors is included in Fig. 15.

4.4 Feedback performance

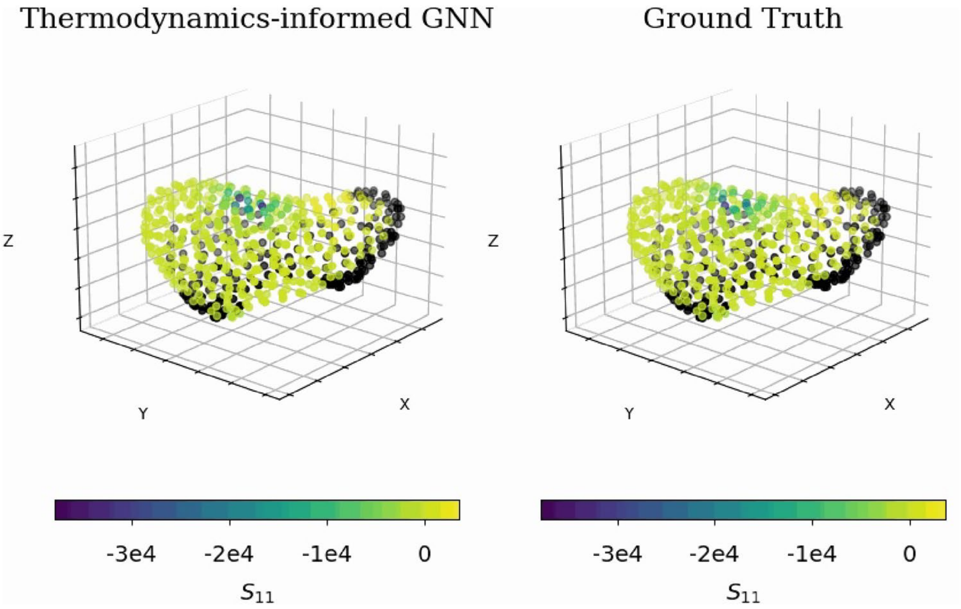
As has already been repeatedly mentioned, the speed of feedback from the haptic peripheral is crucial for realistic sensation. In this section we evaluate the performance of the proposed architectures. The liver model, for instance, consists of a total of 3.8 million parameters, distributed across two hidden layers of 200 neurons each. Its training cost a total of 9 h and 27 min on an Nvidia GTX3090 GPU, across 2000 epochs. Its performance in operation was tested on different architectures to check the influence, especially the type of graphics card, on the results.

When run on a RTX 2070 GPU, the mean elapsed time for a snapshot forecast is 0.011 s (90 Hz). This value decreases to 0.0029 s (344 Hz) in an RTX 4090. Although these figures are generally lower than the 500 Hz often mentioned in the literature, the values obtained are more than sufficient to provide a realistic sensation, both in the haptic gloves and in the Geomagic robot. At no time were there any jumps or stops in the response provided by the gloves, whose main limitation is that they are designed for the manipulation of small objects that fit in the palm of the hand. As the exoskeleton only covers the back of the hand, the sensations due to the movement of the hand as a rigid solid (i.e. no flexing of the fingers) do not produce any type of reaction in the gloves.

5 Conclusions

In this article we have attempted to analyse the validity of artificial intelligence techniques for use as haptic renderers. The restrictions that these systems impose on the response

Fig. 13 Second Piola-Kirchhoff stress field S_x in Pascals at a given snapshot. Comparison between high fidelity, finite element simulation and the learned field



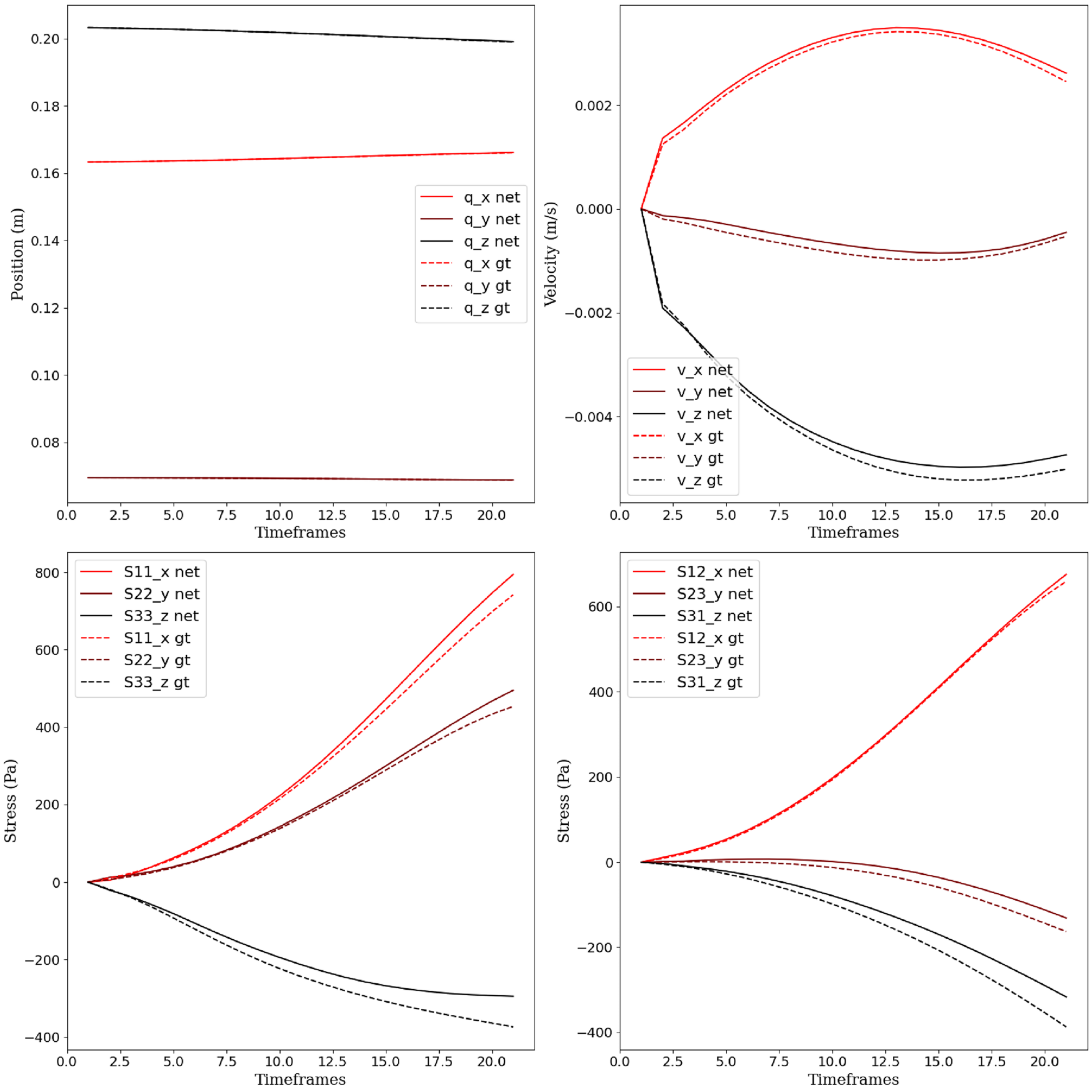


Fig. 14 Comparison between high fidelity, finite element simulation and the learned field for the state variables. From left to right, top to bottom, position, velocity and second Piola-Kirchhoff stress tensor components

speed of the system are well known, because of how severe they are, limiting very much the type of simulation that is used for this purpose.

Graph networks present unique opportunities in this respect. By drawing a parallel between the mesh of high-fidelity models, solved using finite elements, and the graph structure of these networks, we have been able to achieve a very significant degree of realism, at a response speed that, although slightly below 500 Hz, has not allowed us to notice any jump in the transmitted force and, on the contrary, a great realism in the response. This is demonstrated not only by the qualitative, but especially by the quantitative results included in this article.

The highest computational cost of this type of network is in the training process, of course, although this is a once-in-a-lifetime process. Since graph nets actually consist of a series of nets of a very limited size that are shared by all the nodes and edges in the graph, the result is a technique that copes very well with modifications to the mesh and is

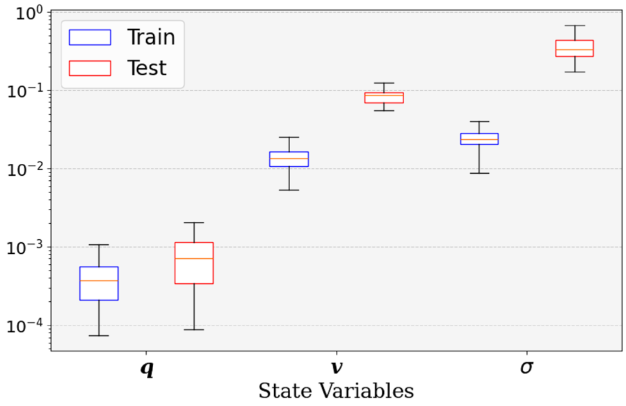


Fig. 15 Relative mean square errors obtained during our experimental campaign for the state variables of the liver problem

very robust to situations not previously seen in the training period.

The trained models have a realistic size that allows us to venture that they can be used in realistic applications, not only in the entertainment industry, but particularly in the industrial design process.

This will open the door to the introduction of haptics synthesis as a tool for industrial design. The feel of commercial products is important, and these kinds of tools make it possible to synthesise realistic responses. More limited, however, is the hardware available. As mentioned, haptic gloves have a fundamental limitation. They are not capable of transmitting any sensation that is not associated with finger flexion. The movement of the hand as a rigid solid does not produce any kind of haptic response in them, severely limiting the realism of some responses. Solving this would require the development of some kind of exoskeleton that would exert force not only on the fingers, but also on the arms and forearms.

Given that some prototypes of these exoskeletons already exist, we are confident that this technology will soon reach the market. At that time, physically consistent haptic rendering engines will be available to achieve a very high degree of realism.

Acknowledgements This work was supported by the Spanish Ministry of Science and Innovation, AEI/10.13039/501100011033, through Grant number TED2021-130105B-I00 and by the Ministry for Digital Transformation and the Civil Service, through the ENIA 2022 Chairs for the creation of university-industry chairs in AI, through Grant TSI-100930-2023-1. This research is also part of the DesCartes programme and is supported by the National Research Foundation, Prime Minister Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. The authors also acknowledge the support of ESI Group through the chairs at the University of Zaragoza and at ENSAM Institute of Technology. Finally, the authors wish to express their gratitude to Miguel Angel Varona and Eduardo Estopiñan for their help in the development of the haptic system.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

Aliabadi FM (2020) Boundary element methods. In: Encyclopedia of continuum mechanics, pp 182–193. Springer

Barbič J, James D (2007) Time-critical distributed contact for 6-dof haptic rendering of adaptively sampled reduced deformable models. In: Proceedings of the 2007 ACM SIGGRAPH/eurographics symposium on computer animation, pp 171–180. Eurographics Association

Belytschko J (2007) A first course in finite elements. Wiley, New York

Chen W, Liu G, Zhang Y, Shirinzadeh B (2016) An artificial neural network based haptic rendering of contact with deformable bodies. In: 2016 IEEE international conference on mechatronics and automation, pp 2332–2337. IEEE

Courtecuisse H, Jung H, Allard J, Duriez C, Lee DY, Cotin S (2010) Gpu-based real-time soft tissue deformation with cutting and haptic feedback. Prog Biophys Mol Biol 103(2):159–168

Deo D, De S (2009) PhyNeSS: a physics-driven neural networks-based surgery simulation system with force feedback. In: Proceedings of 3rd Joint EuroHaptics conference and symposium on haptic interfaces for virtual environment and teleoperator systems, world haptics 2009, pp 30–34. https://doi.org/10.1109/WHC.2009.4810896

Desai S, Mattheakis M, Sondak D, Protopapas P, Roberts S (2021) Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. arXiv preprint arXiv:2107.08024

Escobar-Castillejos D, Noguez J, Neri L, Magana A, Benes B (2016) A review of simulators with haptic devices for medical training. J Med Syst 40:1–22

ESI Group: Press release: enabling a world premiere: 100% of the development of a vehicle done in virtual reality. Accessed on 2024-02-09 (2021). https://www.esi-group.com/news/enabling-a-world-premiere-100-of-the-development-of-a-vehicle-done-in-virtual-reality

Fey M, Lenssen JE (2019) Fast graph representation learning with pytorch geometric. arXiv preprint arXiv:1903.02428

González D, Alfaro I, Quesada C, Cueto E, Chinesta F (2015) Computational vademecums for the real-time simulation of haptic collision between nonlinear solids. Comput Methods Appl Mech Eng 283:210–223

Goyal A, Bengio Y (2022) Inductive biases for deep learning of higher-level cognition. Proc R Soc A 478(2266):20210068

Greydanus SJ, Dzumba M, Yosinski J (2019) Hamiltonian neural networks

Grmela M, Öttinger HC (1997) Dynamics and thermodynamics of complex fluids. I. Development of a general formalism. Phys Rev E 56(6):6620

Heravi N, Culbertson H, Okamura AM, Bohg J (2024) Development and evaluation of a learning-based model for real-time haptic texture rendering. IEEE Trans Haptics 17(4):705–716

Johnsson M, Balkenius C (2007) Neural network models of haptic shape perception. Robot Auton Syst 55(9):720–727

Joolee JB, Jeon S (2021) Data-driven haptic texture modeling and rendering based on deep spatio-temporal networks. IEEE Trans Haptics 15(1):62–67

Keriven N, Peyré G (2019) Universal invariant and equivariant graph neural networks. Adv Neural Inf Process Syst 32:7092–7101

Lang J, Andrews S (2010) Measurement-based modeling of contact forces and textures for haptic rendering. IEEE Trans Vis Comput Graph 17(3):380–391

Laycock SD, Day A (2003) Recent developments and applications of haptic devices. In: Computer graphics forum, vol. 22, pp 117–132. Wiley Online Library

Laycock SD, Day A (2007) A survey of haptic rendering techniques. In: Computer graphics forum, vol. 26, pp 50–65. Wiley Online Library

Maciel A, Halic T, Lu Z, Nedel LP, De S (2009) Using the physx engine for physics-based virtual surgery with force feedback. Int J Med Robot Comput Assisted Surg 5(3):341–353

Martínez Martínez F (2014) Determining the biomechanical behavior of the liver using medical image analysis and evolutionary computation. PhD thesis, Universitat Politècnica de València, Spain

Miller K, Joldes G, Lance D, Wittek A (2007) Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. Commun Numer Methods Eng 23(2):121–134

Niroomandi S, Alfaro I, Cueto E, Chinesta F (2008) Real-time deformable models of non-linear tissues by model reduction techniques. Comput Methods Programs Biomed 91(3):223–231

Niroomandi S, Alfaro I, Gonzalez D, Cueto E, Chinesta F (2011) Real time simulation of surgery by reduced order modelling and x-fem techniques. Int J Numer Methods Biomed Eng 28(5):574–588

Niroomandi S, González D, Alfaro I, Bordeu F, Leygue A, Cueto E, Chinesta F (2013a) Real-time simulation of biological soft tissues: a PGD approach. Int J Numer Methods Biomed Eng 29(5):586–600

Niroomandi S, Alfaro I, González D, Cueto E, Chinesta F (2013b) Model order reduction in hyperelasticity: a proper generalized decomposition approach. Int J Numer Methods Eng 96(3):129–149

Otaduy MA, Okamura A, Subramanian S (2016) Haptic technologies for direct touch in virtual reality. In: ACM SIGGRAPH 2016 Courses, pp 1–123

Otaduy Tristán MA (2004) 6-dof haptic rendering using contact levels of detail and haptic textures. PhD thesis, The University of North Carolina at Chapel Hill, North Carolina

Otaduy MA, Lin MC (2022) High fidelity haptic rendering. Springer, Germany

Öttinger HC, Grmela M (1997) Dynamics and thermodynamics of complex fluids. II. Illustrations of a general formalism. Phys Rev E 56(6):6633

Panait L, Akkary E, Bell RL, Roberts KE, Dudrick SJ, Duffy AJ (2009) The role of haptic feedback in laparoscopic simulation training. J Surg Res 156(2):312–316

Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW (2020) Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409

Quesada C, González D, Alfaro I, Cueto E, Chinesta F (2016) Computational vademecums for real-time simulation of surgical cutting in haptic environments. Int J Numer Methods Eng 108(10):1230–1247

Quesada C, Alfaro I, González D, Chinesta F, Cueto E (2017) Haptic simulation of tissue tearing during surgery. Int J Numer Methods Biomed Eng 34(3):e2926

Quesada C, Badías A, González D, Alfaro I, Chinesta F, Cueto E (2023) Chapter 22 - surgery simulators based on model-order reduction. In: Chinesta F, Cueto E, Payan Y, Ohayon J (eds) Reduced order models for the biomechanics of living organs. biomechanics of living organs, pp 435–452. Academic Press. https://doi.org/10.1016/B978-0-32-389967-3.00029-9. https://www.sciencedirect.com/science/article/pii/B9780323899673000299

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

Rangarajan K, Davis H, Pucher PH (2020) Systematic review of virtual haptics in surgical simulation: A valid educational tool? J Surg Educ 77(2):337–347

Salisbury K, Conti F, Barbagli F (2004) Haptic rendering: introductory concepts. IEEE Comput Graphics Appl 24(2):24–32

Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, Battaglia P (2020) Learning to simulate complex physics with graph networks. In: International conference on machine learning, pp 8459–8468. PMLR

Tesán L, González D, Martins P, Cueto E (2024) Thermodynamics-informed graph neural networks for real-time simulation of digital human twins. arXiv preprint arXiv:2412.12034

Vlachos C, Moustakas K (2024) High–fidelity haptic rendering through implicit neural force representation. In: International conference on human haptic sensing and touch enabled computer applications, pp 493–506. Springer