Optimization in latent space for real-time intraoperative characterization of digital twins

Sidaty El hadramy, Belkacem Acidi, Nicolas Padoy, Stéphane Cotin

Optimization in latent space for real-time intraoperative characterization of digital twins

Sidaty El hadramy[1,2,3*], Belkacem Acidi[4], Nicolas Padoy[2,3], Stéphane Cotin[1,2]

[1]Inria, Strasbourg, France.
[2]University of Strasbourg, CNRS, INSERM, ICube, UMR7357, Strasbourg, France.
[3]IHU Strasbourg, France.
[4]Institut Mines-Telecoms, Paris-Saclay University, Paris, France..

*Corresponding author(s). E-mail(s): sidaty.el-hadramy@inria.fr;
Contributing authors: belkacem.acidi@universite-paris-saclay.fr;
npadoy@unistra.fr; stephane.cotin@inria.fr;

Abstract

Physics-based Digital Twins, particularly those using the finite element method to solve the underlying partial differential equation, accurately simulate organ behaviors but are computationally intensive, especially for hyper-elastic tissues. Recently, approaches have leveraged neural-network-based surrogate models to accelerate computation time. However, these models are limited by the accurate knowledge of patient-specific characteristics, such as material properties and boundary conditions, at training time. This paper introduces a novel methodology for patient-specific characteristics estimation from live observations during medical interventions. To retain the benefits of neural network-based surrogate models, we propose a hypernetwork architecture that conditions the surrogate models on patient-specific characteristics, thus maintaining accuracy over a predefined distribution of these characteristics. Using the trained network, we perform a gradient-based optimization process to determine the patient characteristics given an intraoperative observation. We demonstrate the flexibility and efficiency of our approach through experiments with varying geometries, complex physics laws, and various patient characteristics.

Keywords: Digital Twin, Optimization, Hypernetworks, Patient-specific, Real-Time.

1 Introduction

Digital twins can play a major role in healthcare by leveraging real-time data integration and virtual simulations to enhance patient care, enable predictive analytics, optimize medical devices, and facilitate planning and intraoperative guidance [24]. Several studies have demonstrated the benefits of physics-based digital twins (PBDTs) [9, 15, 5] for their accuracy and predictive ability over sparse and noisy data. PBDTs rely on computational models that simulate the behavior of physical systems using fundamental principles from physics (e.g., elasticity, thermodynamics), where the finite element method (FEM) is often employed to solve the underlying partial differential equations (PDEs) [9]. It is essential to prioritize their accuracy and real-time responsiveness to fully benefit from the advantages of PBDTs in computer-assisted interventions. Accurate models ensure the simulations reflect true-to-life scenarios, leading to more reliable and effective outcomes. Meanwhile, real-time capabilities allow these digital twins to provide immediate feedback and adjustments during interventions, enhancing their practical utility in dynamic and critical environments.

1.1 Accuracy

The accuracy of PBDTs relies on patient-specific characteristics. These characteristics, which are model-dependent, include a range of properties such as density, stiffness, thermal conductivity, and other parameters essential for simulating the behavior of an organ or system [24]. Additionally, boundary conditions (BCs) are critical for the accuracy of PBDTs, as they define the interactions between the organ and its environment, significantly influencing the simulation outcomes. Despite their importance, obtaining precise measurements of these model parameters is challenging. Traditional imaging and other non-invasive techniques often fall short of accurately capturing the necessary details of these properties. For instance, properties like tissue stiffness or thermal conductivity are not readily measurable through standard imaging modalities. Similarly, accurately defining boundary conditions involving complex interactions between tissues and their surrounding environment is often infeasible without invasive procedures.

1.1.1 Material properties

Hyperelastic formulations are commonly employed in the literature to describe the behavior of soft tissues that experience deformations. St. Venant-Kirchhoff, Neo-Hookean [18], and Mooney-Rivlin [22] are among Hyperelastic models that have been used to characterize soft-tissue. Traditionally, parameter sets for various models are determined through uniaxial tests, where researchers identify the parameters that align with the experimentally observed stress-strain relationship according to their strain energy model [26]. This method, while effective, can be limited by its reliance on straightforward experimental setups. To address these limitations, a more advanced approach involving iterative parameter identification using inverse finite element analysis has recently been proposed [1]. This method enhances accuracy by refining parameter estimates through successive approximations. For instance, Mehrabian *et*

al. [14] utilized this technique to estimate tissue parameters modeled with the Veronda-Westmann model. Additionally, Han *et al.* [8] applied this approach to develop a patient-specific biomechanical model of the breast, demonstrating its potential for creating highly personalized and accurate tissue models. However, the iterative nature of these methods means that numerous simulations must be performed to converge on a precise set of patient-specific characteristics. Each simulation can be computationally intensive, especially when modeling complex, nonlinear behaviors typical of soft tissues. Consequently, the time required to reach a solution can hinder the practical application of these methods in dynamic and time-sensitive scenarios, such as surgical procedures or real-time diagnostics.

1.1.2 Boundary conditions

Boundary conditions (BCs) are crucial for the accuracy of PBDTs as they define the interactions between an organ and its surrounding environment. In the literature, zero-displacement boundary conditions are often characterized using priors. For instance, in the case of the liver, the points of attachment with the ligaments are usually considered [15, 5]. However, only a few studies have tackled the challenge of estimating these conditions from intraoperative data. Plantefeve *et al.* [20] used a statistical atlas to estimate liver-ligament connectivity, but their method lacks robustness due to inter-patient variations. Another study estimated BCs by registering two preoperative scans under different deformations. While effective, this approach does not align with clinical settings or real-time requirements. Tagliablue *et al.* [25] proposed a pipeline that estimates BCs from intraoperative point clouds of the visible surface. Nikolaev *et al.* [16] introduced a reduced-order unscented Kalman filter for BC estimation, but their approach is computationally expensive and requires multiple intraoperative samples. These various methods highlight the ongoing challenge of accurately and efficiently estimating boundary conditions from intraoperative data, with each approach offering different trade-offs in terms of robustness and real-time applicability.

1.2 Computational Speed

Significant efforts are being directed toward optimizing PBDTs to enhance their computational speed. Various trade-offs between the speed and accuracy of PBDTs have been proposed [9, 17, 15]. Haouchine *et al.* [9] suggested using a co-rotational model to handle large deformations with small strain. Yet, their approach significantly loses accuracy when advanced biomechanical laws are considered. Depending on the acceptable level of accuracy loss, reducing the model's degrees of freedom is a viable strategy to meet real-time constraints. Methods such as Proper Orthogonal Decomposition (POD) [17] and Proper Generalized Decomposition (PGD) have been proposed for this purpose. Another category of methods leverages the high number of cores available in Graphics Processing Units (GPUs) for parallel computing, which enables significant speedups in handling computationally intensive problems [10]. Deep neural network architectures have recently demonstrated strong capabilities in learning complex, high-level nonlinear relationships between diverse input-output data [11]. One of the strengths of these networks is their ability to perform inference in real-time when

trained with sufficient data accurately. Several works have proposed to train deep neural networks on simulated (using FEM) data [23, 15, 5], aiming to learn the behavior of PBDTs. U-Mesh, introduced by Mendizabal *et al.* [15], stands out as a simple yet effective solution. U-Mesh is a data-driven approach based on a U-Net architecture, designed to approximate the nonlinear relationship between forces and displacement fields. It is trained in a patient-specific manner using simulated data generated by Finite Element Method (FEM) and achieves real-time performance during inference. U-Mesh [15] has shown strong performance on real-world data, making it a promising approach in terms of both accuracy and speed. However, U-Mesh is designed assuming that patient-specific characteristics, such as material properties and domain boundary conditions, are known before the intervention. This assumption limits its application in computer-assisted interventions, as these characteristics, when available, are typically only accessible during the interventions. To address this limitation, we have proposed HyperU-Mesh [3] in previous work, an extension of U-Mesh that integrates a Hypernetwork [7] to condition U-Mesh [15] based on the prior distribution of patient characteristics. Thus, HyperU-Mesh [3] assumes material characteristics are always known at the time of the intervention.

1.3 Contributions and outlines

In this work, we build on HyperU-Mesh [3] and introduce a novel approach to enhance the precision of PBDTs by accurately estimating patient-specific characteristics, including material properties and domain boundary conditions. Our contributions are as follows:

- **Flexible Method:** We develop a method adaptable to unknown variables and dynamically adjusts to the underlying physics model.
- **Hypernetwork Architecture:** We employ a neural network conditioned on patient-specific characteristics through a hypernetwork architecture, allowing the network to adapt to patient characteristics automatically.
- **Training on Simulation Data:** The neural network is trained on simulation data generated using an accurate biomechanical model. This enables it to learn complex relationships between applied forces and the associated displacement fields.
- **Robust and Fast Optimization Process:** After training, given an intraoperative observation, we identify the real patient-specific characteristics through a robust and fast gradient-based optimization process that leverages the neural network's learned representations. The architecture incorporates dimensionality reduction for high-dimensional unknown patient-specific characteristics by optimizing over a latent space representation, thereby accelerating the optimization process.
- **Fast and accurate surrogate model** Once estimated, the patient-specific characteristics are injected through the hypernetwork, allowing for a fast and accurate surrogate model of the PBDT.

After describing our method in Section 2, we illustrate our approach in Section 3 by demonstrating its effectiveness across various scenarios involving different physics models, geometries, and patient characteristics. Finally, we discuss our findings in Section 4 and conclude in Section 5.

2 Method

Without loss of generality, we consider the case where a patient-specific PBDT of an organ is needed to assist in surgery. The geometry of the organ can be segmented from a preoperative CT or MRI scan, while the patient-specific properties, denoted by λ, are unknown. These properties, λ, could include material properties and boundary conditions of the geometry. Let f_λ represent the patient-specific PBDT of the organ for a given set of properties λ. For a set of applied forces to the organ, we have denoted F, the relative displacement field is given by $u = f_\lambda(F)$. Our method consists of three steps. First, we build a biomechanical PBDT by employing the Finite Element Method (FEM), denoted as f_λ^{FEM}. Then, using this biomechanical model, we generate a dataset comprising pairs of forces and displacements, considering a range of distributions for the patient-specific parameters λ. This dataset is used to build a surrogate model, denoted f_λ^{NN}, which corresponds to an approximation of f_λ^{FEM} with a neural network-based model. We use a hypernetwork to condition the neural network on the patient-specific parameters λ. The network is trained on the dataset generated with the biomechanical model. After training, leveraging the gradient flow of the neural network and its real-time capabilities, we use an optimization process to estimate the patient-specific parameters λ based on given observations, denoted y.

2.1 Biomechanical model

We formulate a boundary value problem to compute the deformation of an elastic material under both Dirichlet and Neumann boundary conditions. The geometry of the organ, known *a priori*, occupies a volume Ω with boundary Γ. The Dirichlet and Neumann boundary conditions are on Γ_D and Γ_N, two subsets of Γ. See illustration in Figure 1. The elastic properties of soft tissues can be characterized using principles from continuum mechanics.

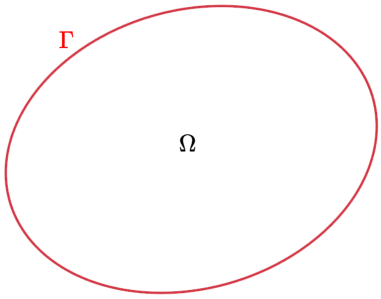

Fig. 1: Illustration of a domain Ω with boundary Γ

By employing the Lagrangian formulation, the relationship between the deformed (x) and undeformed (X) states, at each point along the geometry can be expressed as: $\mathbf{x} = \mathbf{X} + \mathbf{u}$. Where u is the displacement field. The deformation gradient tensor $\boldsymbol{F} = \mathbf{I} + \boldsymbol{\nabla}_{\mathbf{X}} \boldsymbol{u}$, provides a local description of the deformation and the Green-Lagrange

strain tensor $\boldsymbol{E} \in \mathrm{R}^{3\times3}$ as expressed following the equation $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$. Where $\boldsymbol{C} = \boldsymbol{F^T F}$ is called the right Cauchy-Green deformation tensor and $\boldsymbol{I}$ the identity matrix. Hyperelastic material is often employed to describe the behavior of soft tissues that undergo large deformations. According to the chosen hyperelastic material, the strain-energy density function, noted $\mathbf{W}$, can be expressed using a set of parameters that describe the material stiffness. The stress-strain relationship, also known as constitutive law, is obtained by differentiating $\mathbf{W}$ with respect to $\mathbf{C}$ as follows:

$$\mathbf{S} = 2\frac{\partial \mathbf{W}}{\partial \mathbf{C}} \tag{1}$$

With S being the second Piola-Kirchhoff stress tensor. The boundary value problem is then formulated as in equation 2, where g represents the body forces, n the unit normal to Γ_N, and t the traction forces applied on Γ_N domain.

$$\begin{cases} \boldsymbol{\nabla(FS) = g} & \text{on} \quad \Omega \\ \boldsymbol{u(X) = 0} & \text{on} \quad \Gamma_D \\ \boldsymbol{(FS)n = t} & \text{on} \quad \Gamma_N \end{cases} \tag{2}$$

The weak form of Equation 2, brings forward the boundary term and is expressed following Equation 3.

$$\int_\Omega (\boldsymbol{FS}) : \boldsymbol{\delta E}\, d\Omega = \int_\Omega \boldsymbol{b\eta}\, \boldsymbol{d\Omega} + \int_{\Gamma_N} \boldsymbol{t\eta}\, \boldsymbol{d\Gamma} \tag{3}$$

where $\boldsymbol{\delta E} = \frac{1}{2}(\boldsymbol{F^T \nabla \eta + \nabla^T \eta F})$ is the variation of the strain, and $\boldsymbol{\eta} = \{\boldsymbol{\eta} \in \boldsymbol{H^1(\Omega)} \mid \eta = 0 \text{ on } \Gamma_D\}$ is any vector-valued test function in an Hilbert space $\boldsymbol{H^1(\Omega)}$.

2.2 Finite Element solution

For a given set of loads, which could include traction forces t and body force g, the equation 3 is solved using a finite element simulation to retrieve the relative displacement field u, based on a set of patient-specific parameters λ that describe the material properties or the location of Dirichlet boundary conditions. The domain Ω is discretized either using triangle elements or with an Immersed Boundary method (IBM) as shown in the left and right images of Figure 2 respectively. Using an IBM on a regular grid makes it compatible with the inputs and outputs of CNN architectures (See section 2.3). Given the equation's nonlinearity, a nonlinear system of equations must be solved to approximate the unknown displacement. Starting from an initial displacement u^0, an iterative Newton-Raphson method is employed to find a correction δ_u^n after n iterations, which satisfies the linearized set of equations, see Equation 4.

$$\boldsymbol{\dot{K}^{n-1} \delta_u^n = r(u^0 + \delta_u^{n-1}) + b} \tag{4}$$

In this context, $\boldsymbol{\dot{K}}$ represents the tangent stiffness matrix, while $\boldsymbol{r}$ denotes the internal elastic force vector. At each iteration, it is necessary to compute both $\boldsymbol{\dot{K}}$ and $\boldsymbol{r}$ and solve the resulting linear system. The Newton-Raphson method converges effectively only when the initial displacement $\boldsymbol{u^0}$ is close to the actual solution. Therefore, large loads must be applied incrementally in small steps, which can necessitate a significant number of iterations to achieve convergence.

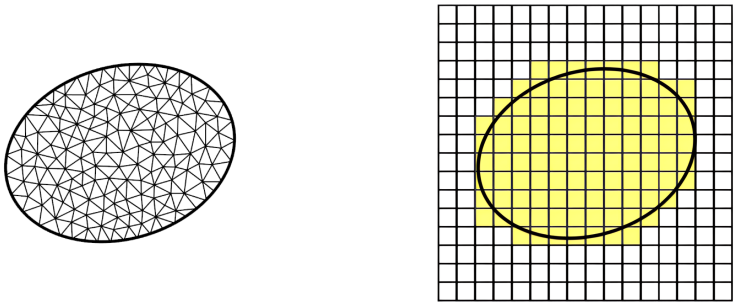

Fig. 2: **Left**: Discretization of the domain Ω using triangle elements. **Right**: Discretization of Ω using an Immersed Boundary Method, the geometry is discretized using regular elements, and the yellow cells are selected as part of the Ω.

2.3 Surrogate model

This section describes the proposed neural network for learning the PBDT, also called the surrogate model. Our approach involves learning a nonlinear function, denoted f_λ^{NN}, to map the applied forces F to the displacement field u with respect to a set of patient-specific parameters λ. To achieve this, we rely on a hypernetwork architecture. Hypernetworks are a class of neural networks that generate the weights of another neural network (usually called target, main, or primary network). They have emerged as a way to enhance the flexibility and performance of deep neural networks [2]. Besides their added adaptability, we are primarily interested in the specific gradient flow of such an architecture since both networks are trained in an end-to-end differentiable manner [7]. Training hypernetworks is often challenging, mainly due to the proportionality between input and output magnitudes, which leads to very slow convergence. Ortiz *et al.* [19] identified and resolved this issue by treating the predicted weights as additive changes for the primary network. For a given training iteration n, the weights $d\theta_n$ predicted by the Hypernetwork h are used to update the weights of the primary network (f_λ^{NN}) following the equation: $\boldsymbol{\theta_n = \theta_0 + d\theta_n}$. With θ_0 the initial weights of the f_λ. Unlike traditional hypernetworks [7], the weights θ_0 are also trainable parameters (See Figure 3). The weights $d\theta_n$ influence the predictions of f_λ^{NN} by incorporating knowledge of the characteristics λ. This strategy permits a better initialization of the primary network weights, which leads to a fast and efficient training of f_λ^{NN}. Typically, the training time of our hypernetwork is comparable to the training time of the primary network on its own.

Figure 3 illustrates the proposed method for learning a PBDT conditioned on patient-specific parameters λ. The dimension of the unknown parameters λ can vary: it may be low, such as for stiffness parameters, or very high, such as when determining the Dirichlet boundary condition location on the organ, in which case λ would have a dimension equal to the number of nodes on the organ's mesh. Depending on the dimensionality of λ, we propose two architectures. For low-dimensional λ (leftmost architecture in Figure 3), it is directly input into the hypernetwork to generate the additive weights for the primary network f_λ^{NN} (the surrogate model). For high-dimensional λ (rightmost architecture in Figure 3), an encoder E is used to reduce the

dimensionality to z. The reduced representation z is then input into the hypernetwork h to generate the additive weights and decoded by D to retrieve the patient parameters λ. The role of dimensionality reduction is significant, as it accelerates the optimization process described in 2.4. The following subsections detail the architecture.

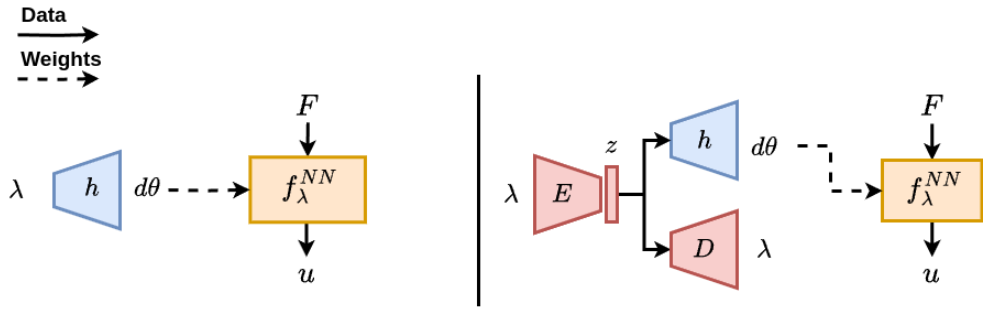

Fig. 3: Overview of the proposed architecture for learning a PBDT conditioned on patient-specific parameters λ. Depending on the dimensionality of λ, we propose two architectures. **Left**: For low-dimensional λ, it is directly input into the hypernetwork h to generate the additive weights for the primary network f_λ^{NN}. **Right**: For high-dimensional λ, an encoder E is used to reduce the dimensionality. The reduced representation z is then input into the hypernetwork h to generate the additive weights and decoded by D to retrieve the patient parameters λ. Reducing the dimensionality of λ speeds up the optimization process for finding λ. In both architectures, the primary network is U-Mesh [15], which takes the applied forces as input and predicts the relative displacement field. h, E and D are multi-layer perceptions.

2.3.1 Primary network

U-Mesh, a Convolutional Neural Network (CNN), proposed by Mendizabal *et al.* [15], has shown strong performance in learning Physics-based biomechanical models on real-world data, making it a promising approach in terms of both accuracy and speed. However, U-Mesh is trained on a single value of material properties and Dirichlet boundary conditions. Hence, its accuracy at inference time depends on precise knowledge of patient-specific parameters during training. In this work, by using U-Mesh as a primary network, we provide the knowledge of the patient-specific parameters λ through the hypernetwork h. U-Mesh takes as input the applied forces as a tensor of size $3 \times n_x \times n_y \times n_z$, where n_x, n_y and n_z are the number of nodes in the FEM mesh along the axes x, y and z, respectively. The forces are encoded at each node as a tensor of dimension 3. The network outputs the relative displacement field of the same size. The encoding and decoding paths of the U-Net consist of 3 and 4 layers, respectively. A layer includes a $3 \times 3 \times 3$ convolution filter followed by a $2 \times 2 \times 2$ max pooling operation and ReLU activation function. The number of channels is $[8, 16, 32]$ and $[32, 16, 8, 3]$ for the encoding and decoding paths respectively.

2.3.2 Other networks

The hypernetwork h is a Multi-layer perception of 4 layers. The last layer is of dimension equal to the number of weights in the U-Mesh, while the first layer's dimension is equal to λ's when λ is for low dimension; otherwise, the input dimension of h is equal to the dimension of λ's latent representation z. A ReLU activation function follows each of h's layers. E and D are also MLPs of four layers each, and they serve as an autoencoder for λ when it has a high dimensionality. A ReLU activation function follows each of their layers. The dimension of the latent space z is experimentally determined based on λ's dimension.

2.3.3 Training

The networks f_λ^{NN}, h, E and D are trained end-to-end and supervised using data generated with the physics-based biomechanical model f_λ^{FEM} explained in Sections 2.1. We utilize an Immersed Boundary Method (IBM) to discretize the domain, resulting in a regular grid, as shown in the right-hand image of Figure 2. This approach is chosen due to its favorable convergence characteristics and regular structure, making it compatible with CNN's architecture chosen for the primary network f_λ^{NN}. Using f_λ^{FEM}, we build a dataset of distinct (F, u, λ) triples, where F represents the applied forces on the organ, u the relative displacement field, and λ the chosen patient-specific parameters. Since λ corresponds to physical quantities (such as soft tissue stiffness or boundary condition location) in our physics-based problem, we often have statistical data on these parameters from the literature. This knowledge is utilized during dataset generation when available, and otherwise, we choose a range of λ values to encompass its variability, thereby capturing diverse patient characteristics. Section 3 presents detailed examples.

The weights of f_λ, E, D and h are optimized by minimizing the loss $\mathcal{L}$ defined in following in Equation 5:

$$\mathcal{L} = ||u - \hat{u}||_2^2 + \mathcal{L}_\lambda \ . \tag{5}$$

In this loss, the first term represents the mean square error between the prediction of the surrogate model $u = f_\lambda^{NN}(F)$ and the ground truth $\hat{u} = f_\lambda^{FEM}(F)$ computed with the biomechanical model. The second term $\mathcal{L}_\lambda$ corresponds to a measure of error between λ and $D(E(\lambda))$ and is included only for the architecture shown in the rightmost image of Figure 3, which applies when λ is of high dimensionality.

2.4 Patient-specific parameter estimation

Once trained, the surrogate model f_λ^{NN} can predict the displacement field of the organ given the applied forces F and the patient-specific parameters λ. However, λ is usually unknown. In this section, we propose using an optimization process that relies on the trained model f_λ^{NN} to estimate the patient-specific parameters λ. Consider a case where the applied forces are known through an appropriate sensor, typically when using surgical robots like the Da Vinci robot. An observation y of the deformation can also be obtained through an intraoperative (imaging) sensor. For instance, laparoscopic

camera, LiDar, ultrasound, or CBCT. The observation y may be full or partial imaging of the organ.

Given the observation y of the deformation, we rely on advanced segmentation and reconstruction techniques [27, 4] to reconstruct the surface of the observation. For instance, point clouds serve as a common representation of any intraoperative observation from the imaging modalities mentioned above. Alongside the applied forces F, the observation y is used to solve an optimization problem to determine the patient-specific parameters λ. This optimization process leverages the trained networks (f_λ^{NN}, h, E and D) and a gradient-based optimizer, such as Stochastic Gradient Descent (SGD), to identify the optimal value λ^* that minimizes an objective function J (described in Equation 6 and 7), hence maximizing the accuracy of the surrogate model f_λ^{NN} on matching the observation y.

$$\lambda^* = \arg\min_\lambda J \tag{6}$$

$$J = \mu(g \circ u, y) \quad \text{with} \quad u = f_\lambda^{NN}(F) \tag{7}$$

In equation 7, g is the preoperative geometry, u the predicted displacement field for a given value of λ and F, y is the point cloud observed intraoperatively, and μ denotes the Euclidean distance between the observed point cloud and the deformed organ. Figure 4 illustrates the Euclidean distance μ between an observation y of the ground truth deformed geometry $\tilde{G}$ and the deformation G computed with the surrogate model when λ is incorrectly estimated. For each point in y, its projection on the surface of G is measured to compute the Euclidean distance.

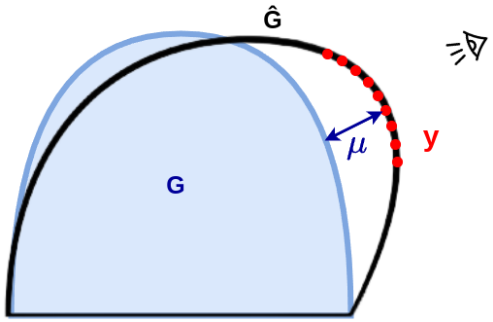

Fig. 4: Illustration of the distance measure μ described in Equation 7. Here, y represents an observation of the deformation $\tilde{G}$, while G denotes the prediction from the surrogate model when λ is misestimated. The objective function is computed as the mean Euclidean distance (μ) between the observed points (y) and the surface of G.

Algorithm 1 shows the optimization process aiming to find the parameters λ that minimizes the objective function in equation 7. We begin by setting an initial guess for λ by sampling from a uniform distribution $\mathcal{P}$, the same distribution used for training. During each iteration, the estimation of λ and the applied forces F are used to predict the displacement field u. This displacement field is used to compute G, a deformation of

Algorithm 1: Optimization Process

Input : The preoperative geometry g, the applied forces F, the intraoperative observation y, and a prior distribution $\mathcal{P}$

Output: An estimation of λ

$\lambda \sim U(\mathcal{P})$
optimizer $\leftarrow$ SGD(λ, lr)
$J \leftarrow$ inf
while $J >$ *threshold* **do**
  $u_\lambda \leftarrow f_\lambda^{NN}(F)$
  $G \leftarrow g \circ u_\lambda$
  $J \leftarrow \mu(G, y)$
  $\lambda \leftarrow$ optimizer(J)
end

the preoperative geometry g. Therefore, the objected function, defined as the Euclidean distance between G and y, is minimized to estimate the optimal λ. The gradient computation is possible thanks to the chain rule described in equation 8 and the automatic differentiation of neural networks. This gradient is the dot product of two terms: (1), which represents the gradient of J with respect to f_λ^{NN} weights, and (2), which corresponds to the gradient of h with respect to its input, note that the first term is identical to the gradient used during training.

$$\frac{\partial J}{\partial \lambda} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial \theta}{\partial \lambda} = \frac{\partial J}{\partial \theta} \cdot \frac{\partial(\theta_0 + d\theta)}{\partial \lambda} = \underbrace{\frac{\partial J}{\partial \theta}}_{(1)} \cdot \underbrace{\frac{\partial(d\theta)}{\partial \lambda}}_{(2)} \tag{8}$$

In equation 8, the initial weights θ_0 of f_λ^{NN} vanish from the second term as they are independent of the parameters λ. This proposed gradient computation remains robust to any initial guess of λ, as gradient computation is consistently feasible due to the automatic differentiation property of neural networks.

When λ has high dimensionality, we use the architecture on the right-hand side of Figure 3. In this scenario, the optimization is performed over the latent-space vector z, which has a lower dimension, thereby reducing the problem's complexity and accelerating the convergence of the optimization algorithm. Upon convergence, the parameter vector $\lambda^* = D(z^*)$ is estimated using the trained decoder D.

3 Experiments and results

To evaluate our method, we conducted two experiments to estimate parameters across various scenarios. All experiments involved implementing the neural network and optimization processes using PyTorch. We utilized the Adam optimizer for the neural network training and the optimization process. The FEM simulations were performed using the SOFA Framework [6] with the SOniCS [13] plugin for soft-tissue biomechanics. Computation was carried out using an Nvidia Titan RTX GPU.

3.1 Experiment 1: Estimation of soft-tissue characteristics

Accurately identifying the material properties is crucial for developing a precise PBDT model. One of the most challenging aspects is determining the stiffness of the tissue, which is often difficult to acquire and can introduce significant errors in physics-based models. This is due to the inherent variability in biological tissues and the complexity of capturing their mechanical behavior accurately. In this first experiment, we aim to estimate the stiffness of an ex-vivo human liver. By accurately determining the liver's stiffness, we aim to improve the fidelity of the PBDT model, leading to better predictions of tissue deformation under various conditions.

3.1.1 Experimental data

An ex-vivo human liver was used for this experiment. Two CT scans were acquired from the liver under two different deformations. As illustrated in Figure 5, the first configuration corresponds to a rest state of the organ, while in the second configuration, a force of 5.1 N was applied on the left lobe. The force was measured with a force sensor and maintained at a constant level during the acquisition by attaching the left lobe to a support. The stiffness of the liver was estimated in average at 7 kPa using the FibroScan® technique. Additionally, a LiDAR camera captured a partial surface point cloud of the organ in the second configuration.

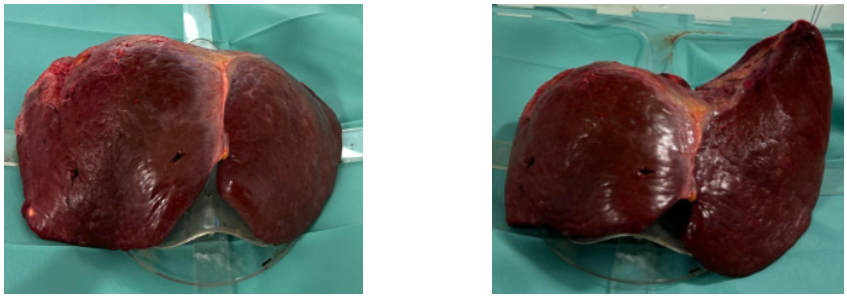

Fig. 5: Rest and deformed states of the ex-vivo human liver.

3.1.2 Biomechanical model ($f_{\boldsymbol{\lambda}}^{\mathbf{FEM}}$)

We follow the boundary value formulation described in Section 2.1 to build $f_{\lambda}^{\mathrm{FEM}}$. The geometry was manually segmented from the CT scan, representing the ex-vivo organ's rest configuration. The liver occupies a volume Ω with boundary Γ. The Dirichlet and Neumann boundary conditions are on Γ_D and Γ_N. Figure 8 illustrates the simulation domain. The Dirichlet boundary conditions are assumed to be known and fixed at the interface between the parenchyma and the portal vein. The material behavior is approximated with a Saint-Venant Kirchhoff model, with the strain-energy density function expressed following Equation 9, with E and υ being respectively the Young Modulus and the Poisson ratio. We recall that $\mathbf{E}$ is the Green-Lagrange strain tensor.

$$\mathbf{W} = \frac{\boldsymbol{Ev}}{\mathbf{(1+v)(1-2v)}}[\mathrm{tr}(\mathbf{E})]^{\mathbf{2}} + \frac{\boldsymbol{E}}{\mathbf{2(1+v)}}\mathrm{tr}(\mathbf{E^2}) \tag{9}$$

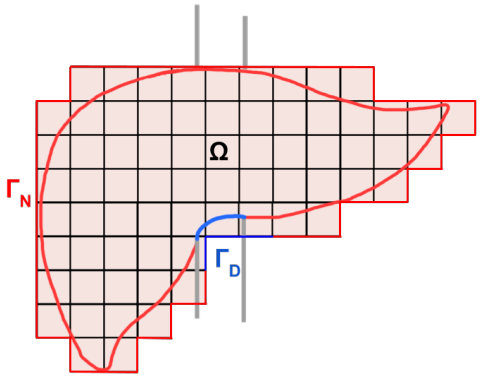

Fig. 6: The liver occupies a volume Ω with boundary Γ. The Dirichlet and Neumann boundary conditions are on Γ_D and Γ_N. We use an IBM to discretize the domain; this is motivated by its compatibility with U-Mesh's CNN architecture.

3.1.3 Data Generation

Using the biomechanical model, denoted f_λ^{FEM}, we generated $3,000$ synthetic deformations of the organ. For each deformation, we randomly selected a location on the liver surface to apply traction forces, sampled from a uniform distribution over [2 N, 10 N]. Moreover, Young's modulus was sampled from a uniform distribution over [5 kPa, 12 kPa], and the Poisson ratio was set at 0.49 for each generated sample. Consequently, a sample from the dataset includes the applied forces F as input for f_λ^{NN}, the chosen Young's modulus as the input λ of the hypernetwork h, and the relative displacement field obtained through f_λ^{FEM}, which serves as the ground truth to supervise f_λ^{NN}'s prediction u.

3.1.4 Network training

In this experiment, the unknown parameter $\lambda = E$ is a scalar representing the material's stiffness. Given the low dimensionality of λ, we use the simpler architecture shown on the left-hand side of Figure 3. The first layer of h is of size 1, h predicts the additive weights $d\theta$ for f_λ^{NN}. These additive weights provide f_λ^{NN} with information about the stiffness. The networks were trained end-to-end on $3,000$ generated samples during 400 epochs with a batch size of 1. This batch size was chosen for its ease in training the hypernetwork. The training optimizer is Adam, with a learning rate of 10^{-4}. The loss function is the mean square error (MSE) between the displacement field predicted by the surrogate model (f_λ^{NN}) and the one computed with the biomechanical model f_λ^{FEM} as expressed in Equation 5. Note that this experiment has no reconstruction loss $\mathcal{L}_\lambda$ as λ is of dimension 1. Hence, no need for dimensionality reduction.

3.1.5 Network results

After training, we evaluated the performance of f_λ^{NN} by applying forces of 5.1 N to the left lobe and comparing the predicted deformation with the ground truth value of the Young's Modulus. We assessed the obtained deformation in terms of Hausdorff distance, and the results are summarized in Table 1. We use the Hausdorff distance because it enables the comparison of meshes (the deformation generated with f_λ^{NN} and the ground truth deformation from Mazier *et al.* [12]) with differing topologies (number of nodes). Additionally, we reported the Relative Hausdorff distance, which represents the percentage error relative to the deformation amplitude. Our results were compared with those from f_λ^{FEM} and U-Mesh [15], where U-Mesh was trained with a constant stiffness value. The comparison shows that the surrogate model, f_λ^{NN}, achieves comparable accuracy to both methods while being faster than f_λ^{FEM} and accommodating a range of stiffness values, unlike U-Mesh. Figure 7 (left image) displays the deformation predicted by the surrogate model f_λ^{NN} superimposed on an image of the actual deformation. To further assess the robustness of our method, we compared the predictions of f_λ^{NN} with the solutions of f_λ^{FEM} in 100 synthetic deformations, yielding a Mean Absolute Error (MAE) of 1.77 mm $\pm$ 0.82 mm. Here, we use a MAE metric as the meshes (deformations generated with f_λ^{NN} and with f_λ^{FEM}) have the same number of nodes (topology). The rightmost image of Figure 7 shows a comparison between the predictions of the surrogate (f_λ^{NN}) and biomechanical (f_λ^{FEM}) models for a given applied forces and a value of λ. The heatmap represents the error at each location in the geometry.

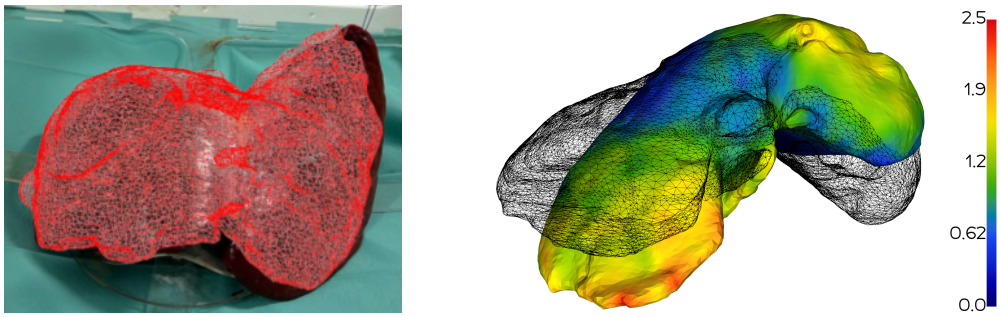

Fig. 7: **Left**: In the wireframe visualization, the prediction of f_λ^{NN} is shown when applying to the rest state (left image of Figure 5) the same loads that produced the deformed state (right image of Figure 5). The predicted result is superimposed over the ground truth deformation image for comparison. **Right**: The wireframe illustrates the liver's rest shape. The surface mesh represents a f_λ^{NN} prediction when specific loads are applied to this rest shape. The heatmap shows the errors of f_λ^{NN} compared to the f_λ^{FEM}'s solution under the same loads.

	Hausdorff (mm)	Relative Hausdorff (%)	Time (ms)
$f_\lambda^{\mathbf{FEM}}$	**15.1**	**11.6**	**500**
U-Mesh	**16.6**	**12.7**	**4**
$f_\lambda^{\mathbf{NN}}$	**16.5**	**12.6**	**4**

Table 1: Results of the liver experiments compare the predictions of standard FEM, U-Mesh, and f_λ^{NN} against the ground truth deformation. f_λ^{NN} achieves results comparable to both state-of-the-art methods while being significantly times faster than standard FEM and more versatile than U-Mesh [15] in handling a range of material properties.

3.1.6 Optimization results

In this section, we evaluate the optimization method for estimating the stiffness of the ex-vivo liver. Given the applied forces at the left lobe, a point cloud of the liver surface was captured using LiDAR. This point cloud is the observation y in the optimization method described in Section 2.4. The Adam optimizer of learning rate of 2.9 efficiently converged after 10 iterations, with an average computation time of 46 ms per iteration. The estimated stiffness value obtained was $\lambda = 6.8$ kPa, leading to an error within 3% of the ground truth value (equals 7 kPa). This result demonstrates the effectiveness of our optimization approach in accurately estimating material properties. The Adam optimizer's efficiency and the estimated value's accuracy underscore our method's robustness in handling real-world data and providing reliable parameter estimation.

3.2 Experiment 2: Boundary condition estimation for soft tissue biomechanics

The boundary conditions (BCs) play a crucial role in the PBDT accuracy as they define the interactions between the organ and its environment. The problem of estimating zero-displacement boundary conditions is challenging since it requires identifying which nodes of the finite element model are constrained. Consequently, the vector λ, over which the optimization takes place, is very large and can contain tens of thousands of unknowns depending on the mesh's size. In such a case, many optimization methods would fail or be time-consuming. In this section, we propose to estimate the zero-displacement boundary conditions using the proposed method.

3.2.1 Experimental data

: For this experiment, we use data from Mazier *et al.* [12]. The dataset consists of a cylinder PolyDiMethylSiloxane (PDMS) beam that deforms under gravity with its left extremity fixed to a vertical support. The non-deformed geometry is provided, and the material behavior is approximated by a nearly incompressible Mooney-Rivlin [12] model with parameters $C_{01} = 101$ kPa and $C_{10} = 151$ kPa. This approximation was performed by Mazier *et al.* using the Mach-1™ mechanical testing system (Biomomentum, Canada). Additionally, an image of the deformation was captured, as illustrated in the left-hand image of Figure 9.

3.2.2 Mechanical model ($f_\lambda^{\mathbf{FEM}}$)

We follow the boundary value formulation in 2.1 to build the biomechanical PBDT, denoted f_λ^{FEM}, for the PDMS beam. The beam occupies a volume Ω with boundary Γ. The Dirichlet and Neumann boundary conditions are on Γ_D and Γ_N, two subsets of Γ. Γ_D is the unknown to be determined. Figure 8 illustrates these domains.

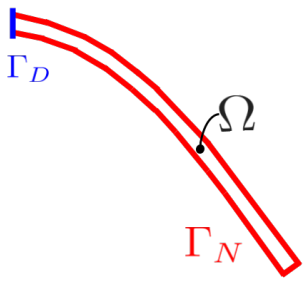

Fig. 8: The beam occupies a volume Ω with boundary Γ. The Dirichlet and Neumann boundary conditions are Γ_D and Γ_N.

The strain-energy density function W for a nearly incompressible Mooney-Rivlin [12] is expressed as:

$$\mathbf{W} = \boldsymbol{C_{01}}(\boldsymbol{J}^{-\frac{2}{3}}\mathbf{I_C} - \mathbf{3}) + \boldsymbol{C_{10}}(\boldsymbol{J}^{-\frac{4}{3}}\mathbf{II_C} - \mathbf{3}) + \frac{\ln(\boldsymbol{J})}{\mathbf{2}\boldsymbol{D_1}} \tag{10}$$

With C_{01}, C_{10} and D_1 being the material parameters, $\mathbf{J}$ is the jacobian matrix, while $\mathbf{I_C} = \text{tr}(\boldsymbol{C})$ and $\mathbf{II_C} = \frac{1}{2}\left((\text{tr}(\mathbf{C}))^2 - \text{tr}\left(\mathbf{C}^2\right)\right)$ are the classic invariants.

3.2.3 Data generation

We generated $6,000$ training samples using the finite element simulation described earlier. For each sample, body forces, denoted F, were selected from a uniform distribution over [7 N, 15 N], and a section of the cylinder was randomly chosen to represent Γ_D. The relative displacement field is $u = f_\lambda^{FEM}(F)$, with λ a discrete representation of Γ_D over the beam's mesh. This representation is a vector where the dimension equals the number of nodes in the FEM mesh (1088 in our case). Each element of the vector is set to 1 if the corresponding node is in the Γ_D domain, and 0 otherwise.

3.2.4 Network training

In the case of this experiment, the parameter λ is a discrete representation of the Dirichlet boundary condition over the beam's mesh, resulting in a high dimensionality of 1088. Therefore, we use the architecture shown on the right-hand side of Figure 3, which includes a dimensionality reduction of λ. The input size of E and the output size of D equal 1088, the latent space representation z having a dimension of 3

	Hausdorff (mm)	Relative Hausdorff (%)	Time (ms)
$f_\lambda^{\mathbf{FEM}}$	**5.8**	**4.4**	**3000**
U-Mesh	**6.2**	**4.7**	**4**
$f_\lambda^{\mathbf{NN}}$	**6.4**	**4.9**	**4**

Table 2: Results from the beam experiments, shows comparisons between predictions made by the biomechanical model (f_λ^{FEM}), U-Mesh, and the surrogate model (f_λ^{NN}) against the ground truth deformation. f_λ^{NN} achieves results comparable to both state-of-the-art methods while operating 750 times faster than the biomechanical model and offering greater flexibility than U-Mesh [15] in handling diverse patient characteristics.

(experimentally chosen). The networks were trained end-to-end during 400 epochs on the $6,000$ generated samples, with a batch size of 1. We used a Binary Cross Entropy for the reconstruction loss of λ, denoted as $\mathcal{L}_\lambda$ in equation 5.

3.2.5 Network results

Aiming for comparison with real data, upon training, we have evaluated the surrogate model (f_λ^{NN}) when the body forces are the gravity and the Dirichlet domain is on the left extremity of the beam. The resulting deformation was compared with the real deformation from Mazier *et al.* [12] and is illustrated in the right-hand image of Figure 9. Similar to the previous experience, we use Hausdorff metrics to compare the predictions and the ground truth geometries. Table 2 presents the numerical results of f_λ^{NN}, and we also report the performances of the biomechanical model f_λ^{FEM} and U-Mesh [15] both compared with the ground truth deformation from Mazier *et al.* [12]. Here, U-Mesh was trained on constant Dirichlet boundary conditions. Results show that f_λ^{NN} achieves comparable results with both f_λ^{FEM} and U-mesh, while operative 750 times faster than f_λ^{FEM} and being more generic than U-Mesh, which is trained for a single value of Dirichlet boundary condition.

To further assess the robustness of f_λ^{NN}, we have compared its results with f_λ^{FEM} over 100 deformations with various locations of Dirichlet boundary conditions. Results achieved a Mean Absolute Error (MAE) of $0.67 \pm 0.57mm$ with a maximum and minimum MAE of $2.7mm$ and $0.12mm$, respectively. We illustrate in Fig. 10 three results, where we compare the prediction of f_λ^{FEM} and f_λ^{NN} on cases where the Dirichlet boundary conditions are fixed on the left, middle, and right cross sections respectively.

3.2.6 Optimization results

An important contribution of this work is the ability to leverage the surrogate model (f_λ^{NN}) to determine the parameters λ that best fit an observation y. In this experiment, we used the image of the deformation shown in Figure 9 (left image) as an observation. Mazier *et al.* [12] reconstructed and scaled the deformed geometry from this image. We used a point cloud of the visible surface as an observation y for the optimization process described in Section 2.4. This point cloud is a constraint for the deformation predicted by f_λ^{NN} during the optimization process. Optimization over z is performed

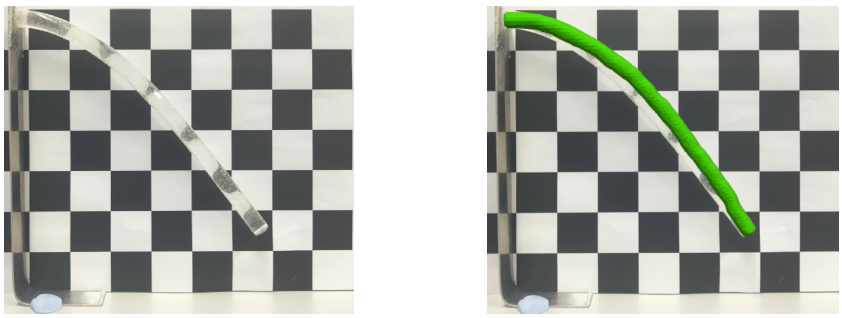

Fig. 9: **Left**: the observed deformation of the beam, fixed on the left side and deforming under gravity. **Right**: prediction of f_λ^{NN} (in green) overlaid onto the ground truth beam.

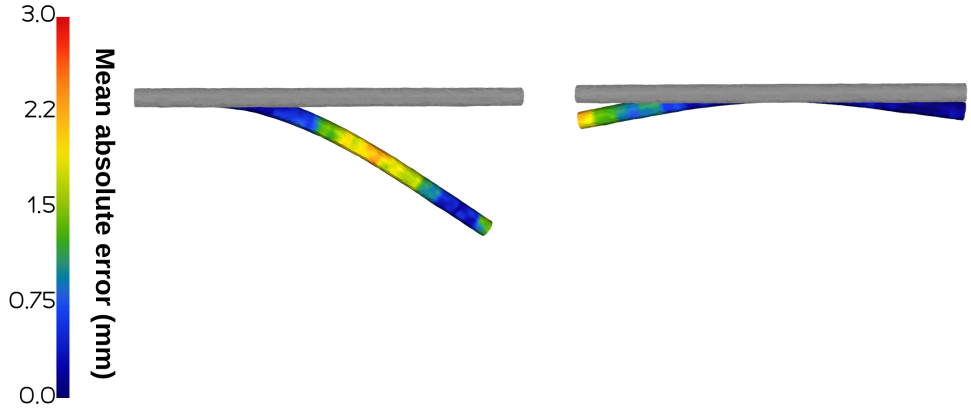

Fig. 10: **Comparison of the surrogate (f_λ^{NN}) and biomechanical (f_λ^{FEM}) models under various deformations and different Dirichlet boundary conditions:** In gray, the non-deformed shape of the geometry, the errors between f_λ^{FEM} and f_λ^{NN} are displayed on the deformed beam. Two examples are illustrated. **Left**: The Dirichlet boundary conditions are on the left extremity. **Right**: The Dirichlet boundary conditions are on a cross-section in the middle of the geometry.

using the Adam optimizer with a learning rate of 2.9 and a random initial guess of z. Convergence was achieved after 10 iterations, resulting in estimations z^* decoded using D to estimate λ. This estimation was input into E and used to predict the deformation of the beam under gravity. Comparison with ground truth resulted in a Hausdorff of 5.8 mm. The optimization over the trained neural network takes, on average, 49 ms per iteration (40 ms for the forward step and 9 ms for the backward step).

3.3 Ablation study

3.3.1 Dimensionality reduction

In cases where the characteristics of the patient λ have a high dimension, we use an autoencoder (which consists of the encoder E and the decoder D) to reduce the dimension to a latent representation z. This approach was applied in Experiment 1. The optimization is then performed on the latent representation z, accelerating the convergence of the optimization algorithm. To justify this, we performed the optimization process in Experiment 2 on the native vector λ without dimensionality reduction, where λ had a dimension of 1088. In this scenario, the optimization algorithm converged after 150 iterations, compared to just 10 iterations when the dimensionality reduction was used. Each iteration takes 49 ms. Thus, the total time for estimating the boundary conditions without dimensionality reduction is 7350 ms, whereas it is only 490ms when optimization is performed on the latent representation.

3.3.2 HyperNetwork architecture

The proposed architecture uses a special type of Hypernetworks, as explained in Section 2.3. Unlike traditional Hypernetworks [7], where the predicted weights completely replace the primary network's weights, our approach uses the predicted weights as additive adjustments to the primary network's weights. This strategy enables faster and more stable training. To support this, we conducted Experiment 1 using both approaches and reported the training losses in Figure 11.

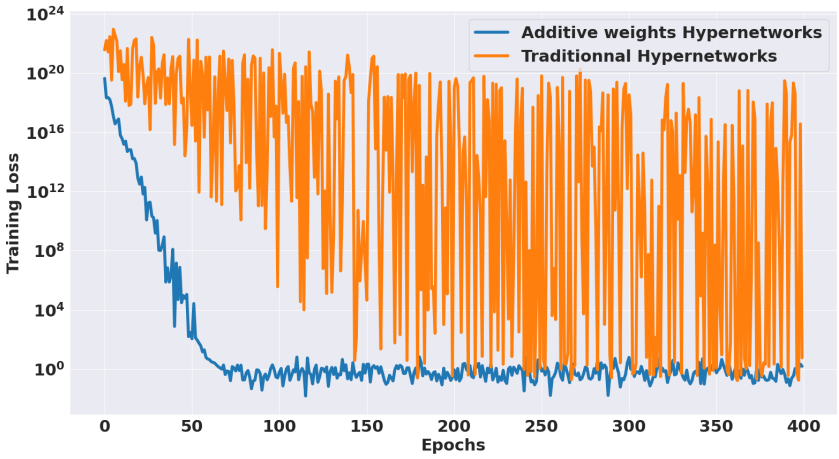

Fig. 11: Comparison of training losses using two hypernetwork strategies. In orange, the traditional hypernetwork strategy is employed, where the hypernetwork fully predicts the primary network's weights. In blue, the hypernetwork predicts additive changes to the primary network's weights.

4 Discussion

Results show that the proposed method is effective in scenarios involving multiple material properties and anatomical structures. The experiments demonstrated that the surrogate model f_λ^{NN} achieves comparable accuracy to both biomechanical model f_λ^{FEM} and U-Mesh while operating significantly faster than the biomechanical model f_λ^{FEM}. Moreover, unlike U-Mesh, which assumes knowledge of patient characteristics before training, f_λ^{NN} demonstrates versatility over a distribution of patient characteristics. This flexibility allows f_λ^{NN} to adapt to patient-specific characteristics intraoperatively, enhancing its robustness and reliability in clinical applications where such variations are prevalent.

The optimization process was effective across both scenarios tested, showcasing its ability to identify patient-specific characteristics that improve the surrogate model's predictive accuracy. Since our primary goal is to ensure the surrogate model's precision, we validated the optimization process based on its ability to find characteristics that yield accurate predictions when used in the surrogate model. This validation confirmed that the proposed approach maintains high accuracy and offers a robust and efficient solution for real-time, patient-specific modeling in clinical settings. Moreover, We utilized point cloud data as intraoperative observations. However, in some scenarios, point clouds may not fully capture the deformation of a 3D geometry, as they only describe the surface of the geometry. We could explore using 3D data from advanced imaging modalities like ultrasound for more comprehensive representation. These in-depth modalities could provide more detailed insights into the internal deformations and characteristics, potentially enhancing the accuracy of the optimization process and, hence, providing more accurate patient characteristics for the surrogate model.

It is worth noting that our architecture is not limited to a specific choice of primary network. While we have chosen U-Mesh for its efficiency, we have also explored other architectures, including Physics-Informed Neural Networks (PINNs) [21], and found them to integrate well as a primary network within the proposed framework. This demonstrates that our method's flexibility extends beyond a specific choice of the primary network, making it adaptable to various neural network architectures suited for different clinical applications. Traditional methods for determining material properties often involve using an optimization process over a biomechanical model to identify the parameters that best fit observed data. In our approach, we replace the biomechanical model with a surrogate model based on a neural network, which significantly accelerates the optimization process. While the biomechanical model itself is only an approximation of the real physical system, the surrogate model serves a similar role. The parameters obtained through the surrogate model do not need to match those derived from the biomechanical model exactly. Instead, they should be the parameters that allow the surrogate model to most accurately represent the real-world behavior.

5 Conclusion

In this work, we present a novel method to improve the accuracy of physics-based digital twins by precisely estimating patient-specific characteristics based on intraoperative observations. Our approach utilizes a hypernetwork architecture to condition

neural network-based surrogate models on patient-specific properties, including material characteristics and boundary conditions. Initially, the network is trained using simulation data from a biomechanical model, accommodating a range of patient characteristics. After training, we employ a gradient-based optimization process to refine the surrogate model, thereby determining the optimal patient-specific parameters that align with the given intraoperative observation. The proposed optimization algorithm converges in real-time, a crucial feature for effective integration into clinical settings. Future work will leverage the hypernetwork to condition surrogate models based on patient anatomical variations. This will enable the development of surrogate models that can adapt in real-time to different patient geometries, material properties, and boundary conditions.

6 Acknowledgments

This work was partially supported by French state funds managed by the ANR under reference ANR-10-IAHU-02 (IHU Strasbourg).

References

[1] Pasu Boonvisut and M Cenk Cavuşoğlu. Estimation of soft tissue mechanical parameters from robotic manipulation data. en. *IEEE ASME Trans. Mechatron.*, 18(5):1602–1611, October 2013.

[2] Vinod Kumar Chauhan, Jiandong Zhou, Ping Lu, Soheila Molaei, and David A Clifton. A brief review of hypernetworks in deep learning, 2023.

[3] Sidaty El Hadramy, Nicolas Padoy, and Stephane Cotin. Hyperu-mesh: real-time deformation of soft-tissues across variable patient-specific parameters. In 2024.

[4] Sidaty El hadramy, Juan Verde, Karl-Philippe Beaudet, Nicolas Padoy, and Stephane Cotin. Trackerless volume reconstruction from intraoperative ultrasound images. In *Lecture Notes in Computer Science*, Lecture notes in computer science, pages 303–312. Springer Nature Switzerland, Cham, 2023.

[5] Sidaty El Hadramy, Juan Verde, Nicolas Padoy, and Stephane Cotin. Towards real-time vessel guided augmented reality for liver surgery. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, 2024.

[6] Francois Faure, Christian Duriez, Herve Delingette, Jeremie Allard, Benjamin Gilles, Stephanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik, and Stephane Cotin. SOFA: a multi-model framework for interactive physical simulation. In *Studies in Mechanobiology, Tissue Engineering and Biomaterials*, pages 283–321. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[7] David Ha, Andrew Dai, and Quoc V Le. HyperNetworks, 2016.

[8] Lianghao Han, John H Hipwell, Christine Tanner, Zeike Taylor, Thomy Mertzanidou, Jorge Cardoso, Sebastien Ourselin, and David J Hawkes. Development of patient-specific biomechanical models for predicting large breast deformation. en. *Phys. Med. Biol.*, 57(2):455–472, January 2012.

[9] Nazim Haouchine, Jeremie Dequidt, Igor Peterlik, Erwan Kerrien, Marie-Odile Berger, and Stephane Cotin. Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Adelaide, Australia. IEEE, 2013.

[10] Stian F Johnsen, Zeike A Taylor, Matthew J Clarkson, John Hipwell, Marc Modat, Bjoern Eiben, Lianghao Han, Yipeng Hu, Thomy Mertzanidou, David J Hawkes, and Sebastien Ourselin. NiftySim: a GPU-based nonlinear finite element package for simulation of soft tissue biomechanics. en. *Int. J. Comput. Assist. Radiol. Surg.*, 10(7):1077–1095, July 2015.

[11] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks*, pages 255–258. MIT Press, London, England, October 1998.

[12] Arnaud Mazier, Alexandre Bilger, Antonio E Forte, Igor Peterlik, Jack S Hale, and Stephane P A Bordas. Inverse deformation analysis: an experimental and numerical assessment using the FEniCS project. en. *Eng. Comput.*, 38(5):4099–4113, October 2022.

[13] Arnaud Mazier, Sidaty El Hadramy, Jean-Nicolas Brunet, Jack S Hale, Stephane Cotin, and Stephane P A Bordas. Sonics: develop intuition on biomechanical systems through interactive error controlled simulations. en. *Eng. Comput.*, 40(3):1857–1876, June 2024.

[14] Hatef Mehrabian and Abbas Samani. Constrained hyperelastic parameters reconstruction of PVA (polyvinyl alcohol) phantom undergoing large deformation. In *Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, Lake Buena Vista, FL. SPIE, February 2009.

[15] Andrea Mendizabal, Pablo Marquez-Neila, and Stephane Cotin. Simulation of hyperelastic materials in real-time using deep learning. en. *Med. Image Anal.*, 59(101569):101569, January 2020.

[16] Sergei Nikolaev and Stephane Cotin. Estimation of boundary conditions for patient-specific liver simulation during augmented surgery. en. *Int. J. Comput. Assist. Radiol. Surg.*, 15(7):1107–1115, July 2020.

[17] S Niroomandi, I Alfaro, E Cueto, and F Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. en. *Comput. Methods Programs Biomed.*:223–231, September 2008.

[18] Nienhuys NW. Phd dissertation. utrecht university; cutting in deformable objects, 2003.

[19] Jose Javier Gonzalez Ortiz, John Guttag, and Adrian Dalca. Magnitude invariant parametrizations improve hypernetwork learning, 2023.

[20] Rosalie Plantefeve, Igor Peterlik, Nazim Haouchine, and Stephane Cotin. Patient-specific biomechanical modeling for guidance during minimally-invasive hepatic surgery. en. *Ann. Biomed. Eng.*, 44(1):139–153, January 2016.

[21] M Raissi, P Perdikaris, and G E Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. en. *J. Comput. Phys.*, 378:686–707, February 2019.

[22] R S Rivlin. Large elastic deformations of isotropic materials IV. further developments of the general theory. en. *Philos. Trans. R. Soc. Lond.*, 241(835):379–397, October 1948.

[23] F Roewer-Despres, Najeeb Khan, and I Stavness. Towards finite-element simulation using deep learning. en, 2018.

[24] Frankangel Servin, Jarrod A Collins, Jon S Heiselman, Katherine C Frederick-Dyer, Virginia B Planz, Sunil K Geevarghese, Daniel B Brown, William R Jarnagin, and Michael I Miga. Simulation of image-guided microwave ablation therapy using a digital twin computational model. en. *IEEE Open J. Eng. Med. Biol.*, 2024.

[25] Eleonora Tagliabue, Marco Piccinelli, Diego Dall'Alba, Juan Verde, Micha Pfeiffer, Riccardo Marin, Stefanie Speidel, Paolo Fiorini, and Stephane Cotin. Intra-operative update of boundary conditions for patient-specific surgical simulation. In *MICCAI 2021*. Springer International Publishing, 2021.

[26] D R Veronda and R A Westmann. Mechanical characterization of skin-finite deformations. en. *J. Biomech.*:111–124, January 1970.

[27] Jakob Wasserthal, Hanns-Christian Breit, Manfred T Meyer, Maurice Pradella, Daniel Hinck, Alexander W Sauter, Tobias Heye, Daniel Boll, Joshy Cyriac, Shan Yang, Michael Bach, and Martin Segeroth. TotalSegmentator: robust segmentation of 104 anatomical structures in CT images, 2022.