Advanced modeling of the childbirth system using different deep learning methods : from fetal skeleton segmentation to real-time soft tissue deformation

Duyen Nguyen

CENTRALE LILLE
THESE

Présentée en vue
d'obtenir le grade de

DOCTEUR

En

Spécialité: Biomécanique et Bioingénierie

Par

Duyen Hien Nguyen-Le

DOCTORAT DELIVRE PAR CENTRALE LILLE

Titre de la thèse:

Modélisation avancée du système d'accouchement à l'aide de différentes méthodes d'apprentissage profond: de la segmentation du squelette fœtal à la déformation des tissus mous en temps réel

Soutenue le 8 Juillet 2024 devant le jury d'examen:

Président	**Marie-Christine HO BA THO,** Professeur, Biomécanique, Université de technologie de Compiègne
Rapportrice	**Valérie BURDIN,** Professeur, Traitement d'images et de signaux, IMT Atlantique
Rapporteur	**Jos VANDER SLOTEN,** Professeur, Biomécanique, KU Leuven
Membre	**Vu-Hieu NGUYEN,** Maître de conférences-HDR, Biomécanique, Université Paris Est Créteil
Directeur de thèse	**Tien-Tuan DAO,** Professeur, Biomécanique, Centrale Lille Institut

Thèse préparée dans le Laboratoire LaMCube
Ecole Doctorale ENGSYS

Acknowledgment

This dissertation marks the completion of my PhD journey, a path filled with both challenges and valuable learning experiences. While I have invested immense effort into this thesis, it also stands as a testament to the support and guidance I received from others and institutions.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Tien-Tuan Dao, whose invaluable guidance, unwavering support, and insightful feedback have been integral to the completion of this dissertation. His expertise and encouragement have been a constant source of motivation throughout my PhD journey.

I would like to acknowledge the Métropole Européenne de Lille organization for their financial support, which made this research possible. I am also very thankful to Centrale Lille Institut and LaMcube laboratory for providing a great working environment that contributed to the success of this journey.

I would like to extend my heartfelt thanks to the members of my dissertation committee, Prof. Valérie Burdin, Prof. Jos Vander Sloten, Assoc. Prof. Vu-Hieu Nguyen, and Prof. Marie-Christine Ho Ba Tho, for their constructive feedback and thoughtful suggestions, which helped shape this research into its final form.

I am incredibly grateful to Dr. Abbass Ballit, who patiently supported me with insightful ideas, and guidance in helping me resolve the problems that emerged during my PhD. His willingness to engage in long discussions was invaluable in addressing the challenges I faced.

I would like to thank my friends and colleagues in LaMcube, Bâtiment ESPRIT. Especially to Thanh Nguyen, Duc-Phong Nguyen, Cong-Ha Nguyen, Victor Lalleman, Xinyue Zhu, Yunlong Wu, and Yushu Wang for their support and friendship along the way.

I am profoundly thankful to my friends and the student community in Lille - chị Ngọc, chị Dung, chị Thanh, chị Phượng, Hoàng Yến, Hoàng Phương, Kiều Oanh, Mai Trang, Thu Hà, Mỹ Nghệ, Hữu Thoại, Việt Hoàng, Đức Mạnh - whom I am truly fortunate to have found. A special thank you to my long-time friends, Khiem Nguyen, Phuc Huynh, and Khai Chau, who have stood by me for many years. Their help, especially with rehearsing my defense, has been significant. I am also deeply grateful to the monks and nuns in the Plum Village, and Ho Quoc Pagoda, who enriched my life with their spiritual gifts, offering peace during challenging times.

My heartfelt gratitude goes to Antoine Lourme for sharing both delight and grief with me, and for taking such good care of me, especially in the weeks leading up to my defense. I could not have done this work without love and encouragement. I would like to express my thanks to Béatrice and Alice for being there on the important day of my defense. I am also sincerely grateful to all the members of the family for bringing joy and helping me learn more about French culture.

Lastly, I wish to extend my gratitude to my friends and family in Vietnam. Living far from home has been a significant challenge, but despite the distance and time zone differences, I am immensely grateful for their encouragement. I am particularly grateful to my mother mẹ Nhơn and brother Dũng, whose love and support made this work possible.

"Walking, eating, breathing, talking, and working are all opportunities to practice creating happiness inside you and around you. Mindful living is an art, and each of us has to train to be an artist."

-Thich Nhat Hanh, How to love

Table of contents

List of Figures

List of Tables

List of Abbreviations

1D	One-dimension
2D	Two-dimension
3D	Three-dimension
4D	Four-dimension
Acc	Accuracy score
AD	Automatic differentiation
Adam	Adaptive moment estimation
Adagrad	Adaptive gradient
AI	Artificial intelligence
ANN	Artificial neural network
BiLSTM	Bidirectional long short-term memory
CNNs	Convolutional neural networks
conv1D	Convolutional neural networks 1D
CPU	Central processing unit
CT	Computed tomography
CycleGAN	Cycle-consistent generative adversarial networks
D	Discriminator
DeepONet	Deep operator networks
Dice	Dice similarity coefficient
DL	Deep learning
dMRI	Dynamic magnetic resonance imaging
DNN	Deep neural network
FEM	Finite element method
FN	False negative
FP	False positive
G	Generator
GAN	Generative adversarial networks
GPU	Graphics processing unit
HPC	High-performance computing
HyperMSM	Hyperelastic Mass-Spring Model
IoU	Intersection over union
L- R-	Left- and right-
LSTM	Long short-term memory
ML	Machine learning
MLP	Multi-layer perceptron
MRE	Magnetic resonance elastography
MRI	Magnetic resonance imaging
MSM	Mass-spring model
MSS	Mass-spring system
NBPP	Neonatal brachial plexus palsy
NeuralODE	Neural ordinary differential equations
NMDID	New Mexico Decedent Image Database
NN	Neural network
NRMSE	Normalized root mean square error
ODEs	Ordinary differential equations

OFF	Object File Format
PCA	Principal component analysis
PCC	Pearson correlation coefficient
PDEs	Partial differential equations
PINN	Physics-informed neural network
PINODE	Physics-informed neural ordinary differential equations
POD	Proper Orthogonal Decomposition
Pointwise CNN	Pointwise convolutional neural network
ReLU	Rectified Linear Unit
RMSE	Root mean square error
RMSprop	Root mean square propagation
RNNs	Recurrent neural networks
ROI	Regions of interest
SD	Standard deviation
SGD	Stochastic gradient descent
SVD	Singular Value Decomposition
TP	True positive
UGGAAN	Unsupervised generative adversarial autoencoding network
WHO	World Health Organization

Chapter 1

Introduction

1.1 Healthcare context

The World Health Organization (WHO) and its partners [1] estimated that the global maternal mortality ratio in 2020 was 223 maternal deaths per 100000 live births. The maternal deaths amounted to 287000 women who died during and following pregnancy and childbirth. This corresponds to almost 800 women dying every day or approximately one woman every two minutes. This is more than a third lower than in 2000 when there were an estimated 446000 maternal deaths. The Sustainable Development Goals target is to reduce the global maternal mortality ratio to less than 70 maternal deaths per 100000 live births by 2030. Maternal mortality is impacted by various factors, among which health system failures play a pivotal role. These failures manifest in several ways, contributing to the challenges faced by pregnant women. First, there are delays in seeking and receiving care after reaching healthcare facilities, exacerbating the risks associated with childbirth. Second, the quality of care provided within these facilities is often subpar, further compromising maternal outcomes. The shortage of essential medical supplies poses an additional barrier, hindering the provision of adequate and timely healthcare services. Moreover, the accountability of the health system is frequently lacking, amplifying the difficulties in addressing maternal health effectively.

According to the WHO report, between 2016 and 2020, pregnancy-related deaths would claim the lives of a woman every two minutes [1], highlighting the persistent issue of maternal mortality. The maternal mortality rate saw an increase in Europe and Northern America, as well as in Latin America and the Caribbean, rising by 17% and 15%, respectively. In contrast, other regions like Australia and New Zealand, along with Central and Southern Asia, significantly decreased maternal mortality rates, with reductions of 35% and 16%, respectively. The immediate causes of maternal death frequently pertain to direct obstetric factors, such as postpartum hemorrhage, which leads to severe blood loss after childbirth. Conditions like pre-eclampsia and hypertensive disorders are also significant contributors to maternal deaths, posing risks due to elevated blood pressure during pregnancy. Furthermore, pregnancy-related infections can develop, causing life-threatening complications if untreated. Maternal fatalities are predominantly concentrated in the world's poorest regions and nations experiencing conflict. In 2020, Sub-Saharan Africa accounted for almost 70% of all maternal fatalities.

In 2019, approximately 2 million stillbirths (i.e., babies delivered with no signs of life at 28 weeks of pregnancy or later) were reported globally [2]. Adequate care could have potentially prevented many of these occurrences. The global stillbirth rate stood at 13.9 stillbirths for every 1000 births. This means that 1 in 72 deliveries results in a fetal death, equivalent to one occurrence every 16 seconds. In the last two decades, the globe has endured a staggering 48 million stillbirths. However, it is important to note that this figure may be underestimated since stillbirths are frequently underreported. The majority of stillbirths can be averted by ensuring access to established interventions that enhance the well-being of both mothers and infants

throughout the care process. It was estimated that 42% of stillbirths worldwide happen during labor. Nearly 832000 deaths in this category in 2019 could have been avoided through the provision of adequate medical treatment during childbirth. This encompasses continuous intrapartum monitoring and prompt intervention in the case of complications. The Sustainable Development Goals aim for every country to achieve a fetal mortality rate of no more than 12 deaths per 1000 live births by the year 2030.

The health of a mother is directly connected to her fetus. In low- and middle-income countries, infections such as malaria, syphilis and HIV rank among the most common causes of stillbirths, accounting for 8% to 50% of all cases [3], [4]. In sub-Saharan Africa, around 20 percent of fetal deaths are linked to malaria, while syphilis is associated with over 11 percent of cases [5]. In all income settings, the mothers' conditions like obesity, diabetes and hypertension are identified as a cause or contributor to stillbirths, accounting for approximately 10% of global cases [6]. Apart from maternal disease, additional variables that may contribute to the danger of both maternal conditions and stillbirths include the mother's age and smoking habits. A study in the United Kingdom revealed that mothers who smoked had a roughly four-fold higher risk of experiencing limitations in fetal growth compared to non-smoking mothers [7]. Furthermore, there are some congenital abnormalities linked to fetal deaths. Due to limits in diagnostic capabilities, the precise percentage of these fatalities in low-income countries is undisclosed. Nevertheless, in high-income nations, they constitute less than 10% of stillbirths recorded nationwide [8]. While congenital anomaly-related fatalities are often considered unavoidable, some can be avoided using readily available therapies like increasing maternal folic acid consumption to lessen neural tube abnormalities.

Intrauterine asphyxia, or a decrease in the fetus's oxygen supply, is the primary factor contributing to the majority of fetal deaths. Placental abruption, protracted labor, preeclampsia, breech presentation (when the feet or buttocks come first) and incidents involving the umbilical cord are some of the variables that may cause this syndrome [9]. The placenta serves as the point of interaction for the transfer of oxygen and nutrients from the mother to the fetus, while simultaneously removing carbon dioxide and other waste products from the fetal circulation. Oxygen-rich blood travels from the placenta to the fetus through the umbilical vein, reaching the fetal liver, where it combines with venous blood from the fetal vena cava. The placenta must operate properly for the best possible fetal growth. If the placenta is unable to protect or nourish the baby, serious problems will ensue [10]. Umbilical cord prolapse represents a critical obstetric emergency where the cord is compressed, leading to the complete or partial blockage of cord vessels and subsequently compromising the oxygen supply to the fetus [11]. The diagnosis of umbilical cord prolapse occurs when the cord has passed past or parallel to the fetal portion that is now visible through the cervix. This situation typically happens following a rupture of the membranes. Fetal fatalities can also result from pregnancy problems, including growth retardation and multiple fetuses.

The robustness of a health system can be determined by looking at the fetal mortality rate, which is a sensitive measure of the standard of treatment throughout pregnancy and delivery. To prevent stillbirths and guarantee that mothers are assisted by qualified healthcare professionals during pregnancy and delivery, nations, international organizations, and partners must take prompt action. Advocating for healthcare accessibility for everyone is crucial for realizing the commitment to universal health coverage and preserving the lives of all children. Additionally, ending preventable maternal mortality and ensuring that women not only survive pregnancy but also thrive in health is essential.

Although knowledge and techniques for reducing maternal deaths and stillbirths have been available for many years, a significant number of women – in various regions worldwide – still face barriers to accessing these life-saving remedies. Climate change and prolonged war, along with overburdened health systems depleted of key supplies and medications, exacerbate the situation for many people. To address this, it is crucial to invest in strengthening our health workforce by providing them with the necessary resources, tools, and training to deliver high-quality treatment that can make a meaningful impact. It is imperative to hold health systems accountable for ensuring the provision of quality, respectful, and equitable care. This can be achieved through a well-trained and supported workforce, as well as ensuring that types of equipment are adequately stocked with essential supplies.

1.2 Background

1.2.1 Overview of childbirth simulations

Childbirth simulations have been investigated in an attempt to forecast and prevent severe complications in both mothers and fetuses during delivery. Constructing a comprehensive model of the pregnant woman's pelvic system and the fetal body is crucial in advancing the understanding of childbirth mechanics. The intricate process involves not only building accurate representations of the maternal pelvic system but also creating detailed models of the fetus inside the uterus. These simulations enable researchers and healthcare professionals to delve into the mechanical properties of pelvic tissues, offering valuable insights into the dynamics of childbirth.

One method used before was statistical analysis of observational data collected in the past. Correlation coefficients between a difficult labor situation and particular maternal/fetal characteristics can be obtained by examining data that describe a wide variety of deliveries. For example, consider the case of cephalopelvic disproportion, which happens when there is an imbalance in the size or shape of the fetal head and the mother's pelvis. This incongruence can lead to failed labor for mechanical causes. If not addressed, this can result in obstructed labor, which potentially elevates the risk of maternal or perinatal death for both the mother and infant unless a cesarean section is performed. Korhonen et al. [12] assessed the diagnostic precision of the ratio of fetal head and maternal pelvis in predicting cephalopelvic disproportion. This study observed 274 pregnant women who underwent X-ray or magnetic resonance imaging pelvimetry from 2000 to 2008 due to an elevated risk of fetal-pelvic disproportion at North Karelia Central Hospital, Finland. Regression methods, both univariate and multivariate, were used to find the cesarean risk variables. Frémondière et al. [13] evaluated the factors that can be used to forecast instrumental aid and cesarean delivery by using discriminant analysis to examine several fetal-pelvic characteristics. The research included the CT scans of 114 pregnant women. There were measurements of 18 foetal and 43 pelvic variables for every mother-fetus pair. Numerous studies have highlighted the limitations of relying solely on radiological pelvimetry to predict the method of delivery [14], [15]. This technique is not possible to anticipate if an infant can successfully traverse the measured pelvis without encountering cephalopelvic disproportion. It merely indicates whether the pelvis is healthy. The fetal head compliance has a confusing effect in the prediction of cephalopelvic disproportion through a straightforward radiological assessment of the interaction between the fetal head and the mother's pelvis. Obstetric professionals often make decisions on the best delivery modes based solely on their intuition and experience if solid tools for anticipating are absent. This

reliance can lead to numerous women facing challenges in naturally giving birth, and a considerable number of cesarean sections are probably carried out unnecessarily.

Utilizing physical birth simulators presents an alternative method for comprehending the childbirth process. These simulators enable healthcare professionals to gain practical experience in delivery training, which is an important aspect of preserving the lives of both mothers and their infants. Laerdal Medical Health Stavanger has created the MamaNatalie learning kit, which is a low-tech simulator to train medical students on how to manage and detect delivery situations [16] (Figure 1). This birth simulator is a user-friendly pregnant belly that instructors can easily strap on. Using only their hands, instructors can simulate various birth scenarios, including fetal heart sounds, placenta delivery, baby positions in the birth tract, and uterine firmness. MamaNatalie offers a training device for replicating the leading cause of maternal mortality during childbirth is postpartum hemorrhage. When wearing MamaNatalie, the instructor assumes the position of the mother, controlling the amount of blood lost and the rate at which it occurs. They may even mimic a mother who is experiencing shock or extreme blood loss. In addition, teaching the students how to effectively interact with the mother and teammates in such a crucial and traumatic scenario. Gaumard Scientific, based in Coral Gables, has developed a high-fidelity childbirth patient simulator named Noelle, designed to enhance training in labor and delivery, postpartum hemorrhage, and neonatal resuscitation [17] (Figure 2). This full-sized female birth simulator offers a comprehensive teaching system to provide a complete birthing simulation experience, covering care before, during, and after delivery. The set includes a full-sized intubatable manikin, an articulated birthing baby, and a full-sized, intubatable, and cyanotic newborn that changes color with positive pressure ventilation. Notable features of the simulator include a removable abdominal cover, articulated arms, multiple fetal heart sounds, and various placenta locations. This innovative simulation tool is tailored to equip healthcare professionals with hands-on experience in managing scenarios associated with childbirth.

Figure 1. MamaNatalie childbirth simulator includes a blood tank, uterus, placenta, umbilical cord, and infant [16].

Figure 2. Noelle full-sized female birth simulator with fetuses (https://www.gaumard.com/).

However, physical birth models have several drawbacks when it comes to providing a more in-depth understanding of the mechanisms behind birth-related injuries. Initially, the capacity of these simulators to depict patient-specific data is restricted. Typically, they are designed to imitate a typical delivery scenario, which includes an average-sized woman and fetus. It is difficult to modify a simulator to fit measurements unique to a patient once it is built. Secondly, there is a limit to how complicated physical delivery models can capture all the aspects that are essential to the biomechanics of giving birth. Physical delivery models tend to oversimplify soft tissue geometry, such as the muscles of the pelvic floor, without attempting to validate the mechanical characteristics of the soft tissue. Accurately simulating complex physiology on physical simulators becomes problematic.

Childbirth computational models have been explored since the 1990s using numerical methods to investigate how childbirth affects the tissues of mothers and newborns [18], [19], [20]. The approaches have the potential to address certain constraints in physical childbirth models, providing a more profound understanding of the mechanisms underlying maternal and fetal injuries during labor and delivery. Yan et al. [21] explored the impact of fetal head shape variation on the second stage of labor with particular attention to potential problems and consequences for the mother and fetus. The second stage of labor, marked by the descent of the baby through the birth canal, is crucial for a successful delivery. To analyze this influence, the study employed the finite element method. Childbirth simulations were performed with skull models, which were generated from computed tomography (CT) images of fetal heads, and female pelvic floor models based on magnetic resonance imaging (MRI). Krofta et al. [22] developed MRI-based model to simulate the deformation of the levator ani muscle during vaginal delivery. The stress experienced during childbirth is known to potentially result in stretch-related injuries to the levator ani muscle. A detailed 3D finite element model of the maternal pelvis and fetal head was created using MRI images of newborn fetuses and nulliparous women for geometry. A virtual childbirth simulation tool utilizing Mixed-Reality (MR) technology and integrating Hyperelastic Mass-Spring Model (HyperMSM) for real-time soft tissue displacements was proposed by Ballit et al. [23] (Figure 3). This innovation aimed to improve the midwife's skills through user interaction with the virtual physical model. The Microsoft HoloLens was employed in developing the MR simulator, which features a comprehensive holographic obstetric model. The uterus, the fetus, the birth canal, the pelvis bone, and the pelvic floor muscles were all included in the maternal pelvis system model of a pregnant woman.

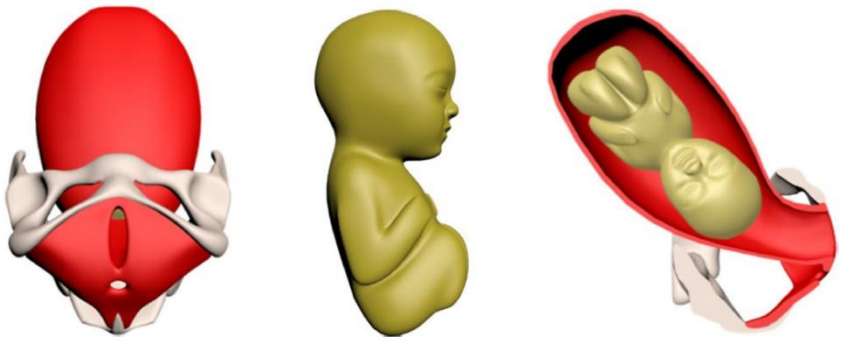


Figure 3. General geometric representations of the maternal pelvic system include rigid and flexible tissues along with a fetus [23].

The development of computer technology and numerical modeling software has led to the creation of more realistic and advanced computational models for birthing, capable of examining intricate subjects. However, current studies on childbirth simulations face some limitations. The primary focus of research has been on pelvic floor injuries sustained by mothers during delivery, with the popular portrayed maternal pelvic system being the levator ani muscles. Fetal models, in many instances, have been limited to including only the head, with joint articulations infrequently considered. The delivery process has often been driven by pre-determined forces. Moreover, materials with extremely high stiffness have been used to depict fetal models.

1.2.2 Applications of Machine learning and Deep learning in Healthcare

1.2.2.1 Machine learning and Deep learning

Artificial intelligence (AI) has grown to be a vital component of our daily lives, exerting a significant influence on society nowadays. Thanks to the expansion of computational capabilities, advancements in methodologies and the increase of data volume, AI has established itself as a crucial technological asset across a wide range of industry, business, entertainment and education [24], [25], [26], [27], [28], [29]. Machine learning, a subfield of AI, is dedicated to creating models and algorithms that enable computers to learn from information and enhance their performance based on prior experiences. In simple terms, machine learning allows systems to emulate human-like thinking and understanding by learning from data. Deep learning, a subfield of machine learning, employs neural networks to tackle complicated problems. Neural networks are inspired by the structure and operations of the human brain, comprising layers of linked nodes for processing and manipulating data.

An artificial neural network (ANN) consists of a series of equations designed to mimic the inherent connections within a provided dataset. By examining samples, an ANN discovers patterns in the data, typically operating without explicit and task-specific instructions. The basic structure of an ANN is illustrated as in Figure 4.

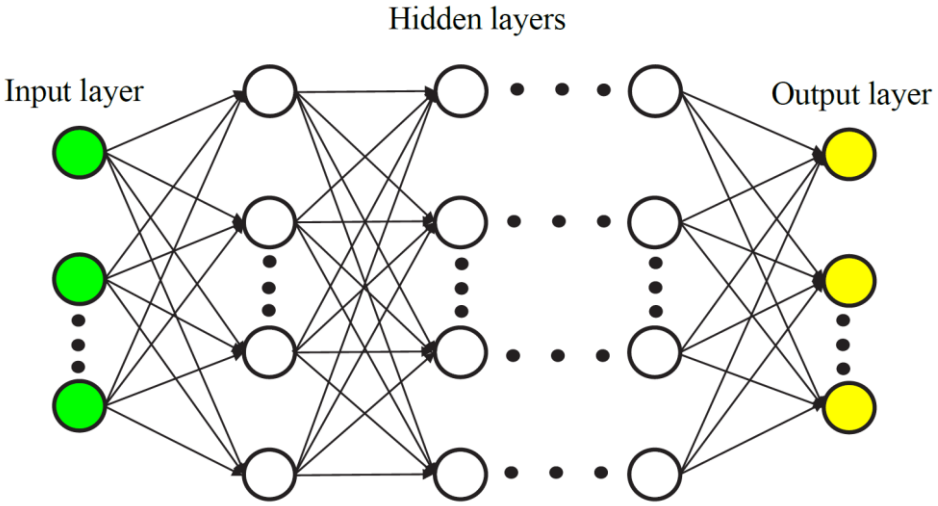


Figure 4. Artificial neural network architecture.

Giving the input dataset $X = \{x_1, x_2, \dots, x_m\}$, weight set for the *i*-th hidden neuron node $W^i = \{w_1^i, w_2^i, \dots, w_m^i\}$ and bias parameters $b^i = \{b_1^i, b_2^i, \dots, b_m^i\}$, when applying the dot produce summation along with the bias, a result z is obtained as follows

$$z^i = \sum_{j=1}^{m} w_j^i x_j + b_j^i \tag{1}$$

To calculate the values of the hidden layer, we feed z into an activation function $f(.)$. Activation functions play a pivotal role in neural networks as they introduce non-linearity, determining which nodes will be activated. Some common activation functions include:

- Identity function

$$f(x) = x \tag{2}$$

- Sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

- Tanh function

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4}$$

- Rectified Linear Unit (ReLU)

$$f(x) = \begin{cases} 0 & for\ x < 0 \\ x & for\ x \geq 0 \end{cases} \tag{5}$$

- SoftPlus function

$$f(x) = ln(1 + e^x) \tag{6}$$

The value of hidden node *i*-th is given by

$$h^i = f(z^i) \tag{7}$$

After hidden layers, having n values of hidden nodes $h = \{h_1, h_2, \dots, h_n\}$, a weighted set $W^h = \{w_1^h, w_2^h, \dots, w_n^h\}$ and a bias set $b^h = \{b_1^h, b_2^h, \dots, b_n^h\}$, the output $\hat{y}$ is obtained as follows

$$z = \sum_{j=1}^{n} w_j^h h_j + b_j^h \tag{8}$$

$$\hat{y} = f(z) \tag{9}$$

During the training process, the loss function is minimized to assess the network's performance on the training dataset, allowing it to adjust and navigate appropriately. The predominant loss function employed is the mean square error, which measures the disparity between k predicted values $\hat{y}$ and k target values y as below

$$\mathcal{L} = \frac{1}{k} \sum_{j=1}^{k} (\hat{y}_j - y_j)^2 \tag{10}$$

As the loss function $\mathcal{L}$ relies on weighted and bias parameters, optimization functions are invoked to adjust these parameters in order to minimize it. Gradient-based algorithms are commonly employed for training neural networks with several prevalent optimizers including:

- Stochastic gradient descent (SGD) directly updates and returns weights and biases in each time step t by

$$\theta_{t+1,i} = \theta_{t,i} - \eta \cdot g_{t,i} \nabla \mathcal{L}(\theta) \tag{11}$$

where θ denotes parameters, η denotes learning rate and $g_{t,i} = \nabla \mathcal{L}(\theta_{t,i})$ is the partial derivative of the loss function $\mathcal{L}$.

- Adaptive gradient (Adagrad) algorithm [30] adapts the learning rate of each parameter during training, scaling it based on the previous gradients for that parameter

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \varepsilon}} \cdot g_{t,i} \tag{12}$$

in which G_t stands for a diagonal matrix where each diagonal element i and ε is a smoothing term to avoid divisibility by.

- Adadelta [31] is a variation of the Adagrad optimizer. Instead of accumulating all past squared gradients, this algorithm restricts the window of accumulated past gradients to a fixed-size sliding window to avoid the diminishing learning rate encountered in Adagrad

$$\theta_{t+1} = \theta_t - \frac{RMSE[\Delta\theta]_{t-1}}{RMSE[g]_t} \cdot g_t \tag{13}$$

where $RMSE$ is the root mean square error.

- Root mean square propagation (RMSprop) [32] is a method that addresses Adagrad's declining learning rates. It keeps a moving average of squared gradients that decays over time, providing greater weight to recent gradients and thereby preventing the learning rate from falling too low too rapidly

$$\theta_{t+1} = \theta_{t,i} - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} \cdot g_t \tag{14}$$

where $E[g^2]_t$ is a running average at the current time step, defined as:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \tag{15}$$

- Adaptive moment estimation (Adam) [33] is an optimizer commonly used for training neural networks. Adam combines advantages of both Adagrad and RMSprop. The algorithm maintains two moving averages, the first and second moment of the gradients, which are expressed as:

$$m_t = \beta_1 m_t + (1 - \beta_1) g_t \tag{16}$$

$$v_t = \beta_2 v_t + (1 - \beta_2) g_t^2 \tag{17}$$

where $\beta_1, \beta_2 \in [0,1]$ manage the exponential decay rates of the moving averages. Subsequently, calculate the bias-corrected first and second moment estimates

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{18}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{19}$$

The update rule for Adam optimizer is as stated below:

$$\theta_{t+1} = \theta_{t,i} - \frac{\eta}{\sqrt{\widehat{v}_t + \varepsilon}} \widehat{m}_t \tag{20}$$

Machine learning can be viewed as a practical application of statistics with a heightened reliance on computer assistance to estimate complex functions statistically while placing less demonstration on establishing ranges of confidence across these functions. The majority of machine learning algorithms can be categorized into supervised learning, unsupervised learning and reinforcement learning methodologies. The categorization is based on the type of experience they are permitted to undergo throughout the learning process.

Supervised learning methods operate with a feature-rich dataset where each sample is linked to a label or target. Models are trained on datasets consisting of both inputs and corresponding outputs. They are then employed to forecast outcomes on test datasets, wherein only the inputs are given. The outputs generated by the model are compared against the concealed target variables, thereby gauging the model's proficiency. Two primary categories of supervised learning problems exist: classification, which entails forecasting a class label, and regression, which includes predicting a numerical value.

Unsupervised learning methods operate a dataset with numerous features, from which they extract valuable characteristics regarding the dataset's architecture. Models aim to learn the whole probability distribution responsible for generating a dataset, whether through implicit methods for duties such as synthesis and denoising, or explicit approaches such as densities

calculation. In contrast to supervised learning, unsupervised learning solely relies on inputs without corresponding outputs or targets. Thus, unsupervised learning lacks an instructor to modify the model. There are two major issues that users frequently face: clustering, which includes identifying groups within data, and density estimation, which requires summarizing the data's distribution.

Reinforcement learning methods operate in an environment where there is no preset training dataset. Instead, there exist specific objectives for an agent to accomplish, and feedback provided regarding its performance in reaching those objectives. Similar to supervised learning, in reinforcement learning, the model receives feedback to learn from. However, reinforcement learning might be difficult for the model in establishing causal relationships because the feedback could be delayed and contain statistical noise. There are two primary types into which reinforcement learning can be classified: positive reinforcement, which involves rewarding the agent for executing a desired action, thereby motivating it to repeat the behavior, and negative reinforcement, which entails removing an undesired stimulus to motivate a desired activity.

Deep learning methods have a main feature that is the usage of deep neural networks, which consist of numerous layers of interconnected nodes. These networks possess the ability to grasp intricate data representations by identifying structured patterns and characteristics. Deep learning models can autonomously learn and enhance themselves through data, eliminating the requirement of manual adjusting. Some of the most noticeable deep learning architectures contain recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

Recurrent neural networks are a family of neural networks tailored for handling sequential data. RNNs excel in managing significantly longer sequences compared to non-sequence-specialized networks, making them feasible for practical applications. Additionally, most recurrent networks are adept at processing sequences of varying lengths. RNNs use previous inputs to shape the present input and outcome. Unlike traditional deep neural networks, which assume independence between inputs and outputs, RNNs base their outputs on preceding items in the sequence. While considering future occurrences could enhance output prediction, unidirectional RNNs lack the ability to incorporate such events. Variant RNNs structures encompass Bidirectional recurrent neural networks [34], Long short-term memory [35] and Gated recurrent units [36].

Convolutional neural networks are a specific type of deep learning that processes data using predefined grid-like structures, predominantly employed in tasks such as classification and computer vision. CNNs signify the utilization of convolution, a specific type of linear operation. In essence, CNNs are neural networks that substitute convolution for matrix multiplication in at least one layer, distinguishing them from traditional networks. CNNs consist of three primary types of layers: *the convolutional layer*, which serves as the fundamental component of a CNN and is responsible for the majority of computational tasks, requiring input data, a filter, and generating a feature map. *The pooling layer*, also referred to as downsampling, diminishes dimensionality, reducing input parameters. Unlike the convolutional layer, the pooling operation applies a filter across the entire input without any weights, employing an aggregation function within the receptive field to populate the output array. Finally, *the fully connected layer* connects each node in the output layer directly to a node in the preceding layer, facilitating classification based on features extracted from earlier layers and their respective filters. There are some popular convolutional neural network architectures such as Residual Networks [37] and U-Net [38].

1.2.2.2 Clinical care applications

In recent years, AI has undergone remarkable advancements, making it a compelling tool for various clinical care procedures. Its potential applications in medicine have garnered increasing research attention, with Machine learning (ML) and Deep learning (DL) emerging as indispensable technologies across the medical field. From diagnostics to therapeutics, prognosis prediction, and patient health management, AI is revolutionizing healthcare delivery. Moreover, it plays a vital role in hospital administration, as well as in regulatory activities within healthcare systems.

Machine learning and Deep learning have applied well in conducting various diagnostic tasks, encompassing omics and clinical data processing. Within cardiovascular medicine, numerous models have been developed for predicting and diagnosing various conditions such as incident hypertension [39], detecting atrial fibrillation [40], and classifying aortic stenosis [41]. Similarly, in clinical neurosciences, tools have been devised for predicting strokes [42], [43]. In hematology, these technologies have significantly advanced the diagnosis of conditions such as lymphoma, leukemia, and thalassemia [44], [45], [46]. ML and DL technologies are particularly adept at medical image detection. Many segmentation tasks, including right and left ventricular segmentation and atrial wall segmentation [47], [48], are effectively handled by ML and DL techniques. Moreover, these technologies excel in segmenting organs such as the liver, kidney, bladder, vagina, and rectum [49], [50], [51], as well as in detecting cancerous tumors [52], [53]. Google DeepMind [54] has developed a groundbreaking predictive model capable of foreseeing kidney injury up to 48 hours in advance with an astonishing accuracy rate of 90%. This remarkable capability provides medical professionals with crucial lead time for preemptive interventions, potentially saving lives and improving patient outcomes. Similarly, Zebra Medical Vision [55] has pioneered algorithms adept at detecting signs of various diseases and conditions within medical images, spanning from lung cancer and breast cancer to liver diseases and cardiovascular conditions. By leveraging these AI-driven tools, radiologists can enhance their diagnostic accuracy and expedite the detection of illnesses in their early stages, leading to more effective treatment strategies and improved patient care. Furthermore, ML and DL play a crucial role in identifying diseases such as dementia and Alzheimer's [56], Parkinson's [57], multiple sclerosis [58], and brain abnormalities in temporal lobe epilepsy patients [59].

Utilizing artificial intelligence methods in drug development offers a comprehensive approach to understanding molecular properties. Through *in silico* modeling, these methods not only aid in determining the presence of lead compounds with optimal properties but also offer theoretical insights into absorption, bioactivity, metabolism, side effects, and excretion of the drug. Moreover, by using multi-objective optimization approaches, AI assists in lowering attrition costs throughout the preclinical phase. AI has proven its effectiveness in a number of areas, including estimating drug-protein interactions [60], determining medication efficacy [61], and guaranteeing biomarker security [62]. Cutting-edge ML and DL models that predict protein structures like AlphaFold [63], and tools that evaluate protein quality such as ProQ3D [64], QACon [65] and DeepQA [66], showcase the breadth of AI influence in advancing drug development and therapeutics. OrthoGrid Hip [67] is advanced software designed to correct distortion and ensure precise alignment during direct anterior total hip arthroplasty. It is compatible with various implants and operates on an AI-powered digital platform.

Additionally, artificial intelligence is reshaping patient health management through innovative technologies. Wearable devices powered by AI enable continuous collection and monitoring of vital patient data, including heart rate, blood pressure and sleep patterns. This real-time information serves as a valuable tool for both individuals and healthcare professionals, facilitating the tracking of health trends and early detection of anomalies. AI virtual assistants further enhance patient care by assisting with appointment scheduling, medication reminders, and providing basic medical information in a conversational manner. These virtual agents play a progressive role in patient wellness, offering personalized recommendations on diet, exercise, and stress reduction. A notable example of AI in healthcare is IBM Watson Health [68], which harnesses AI, machine learning, and natural language processing to analyze extensive medical data. By assisting healthcare professionals in decision-making, IBM Watson Health empowers more informed and effective patient care strategies.

The rapid integration of AI into healthcare practices signifies a paradigm shift in how medical services are delivered and regulated, promising significant improvements in patient outcomes and overall healthcare efficiency.

1.3 Objectives

Childbirth simulations have been investigated in an attempt to forecast and prevent severe complications in both mothers and fetuses during delivery. Constructing a comprehensive model of the pregnant woman's pelvic system and the fetal body is crucial in advancing the understanding of childbirth mechanics. However, scientific challenges remain in the realistic representation of the fetus and suitable computational cost and processing speed to deploy the childbirth simulations into clinical routine practices.

This PhD thesis has three original contributions to overcome these challenges:

1) Automated segmentation of fetal skeletons into distinct components utilizing a model based on generative adversarial networks (GAN) and 3D point cloud data.

2) Forecasting real-time soft tissue deformations employing recurrent neural networks (i.e. long short-term memory neural networks (LSTM)) integrated with a learning strategy based on principal component analysis (PCA).

3) Designing and assessing an outstanding model for simulating real-time soft tissue deformations, leveraging the physics-informed Neural Networks (PINN) and Neural Ordinary Differential Equations (NeuralODE).

This thesis introduces novel methods for accurately modeling fetal representation and dynamically simulating soft tissue deformation in real-time. These advancements aim to contribute to the development of a next-generation decision support tool tailored for childbirth training and simulating complications. By combining innovative techniques, this research lays the foundation for more realistic simulations that can enhance training outcomes and facilitate a better understanding of childbirth complexities.

1.4 Thesis organization

The rest of the thesis is organized as follows:

- **Chapter 2:** *State-of-the-Art*. The chapter reviews and explores the state-of-the-art of newborn 3D skeleton part segmentation and the predictions of soft tissue deformations in real-time.
- **Chapter 3:** *Generative adversarial network for newborn 3D skeleton part segmentation*. We propose a generative adversarial network-based model to segment the fetal skeleton into separate parts using 3D point cloud data.
- **Chapter 4:** *A novel deep learning-driven approach for predicting the pelvis soft-tissue deformations toward a real-time interactive childbirth simulation.* Deep learning-based models to predict the deformation of pelvic soft tissue within interactive childbirth simulators are built and evaluated. This involves the integration of long short-term memory neural networks (LSTM), deep neural networks and principal component analysis (PCA)-based learning.
- **Chapter 5:** *Physics-informed neural ordinary differential equations and mass-spring system modeling framework for predicting the real-time soft-tissue deformations.* We develop and evaluate the state-of-the-art model to simulate real-time deformations of soft tissue. This innovative approach employs Physics-informed Neural Networks (PINN) and Neural Ordinary Differential Equations (NeuralODE).
- **Chapter 6:** *General discussion.* The key contributions and a summary of the work will be discussed.
- **Chapter 7:** *Conclusions and perspectives.* The final chapter presents conclusions and perspectives.

Chapter 2

State-of-the-Art

2.1 3D part segmentation

2.1.1 Summary of used methods

Deep learning has been intensively used to perform medical image analysis tasks such as classification, reconstruction, and segmentation. This advancement has paved the way for more accurate and efficient analysis of complex medical data, leading to improved diagnostic accuracy and patient care. Segmentation of female pelvis organs is essential for a wide range of clinical applications, including gynecology, oncology, and obstetrics. Accurate delineation of organs such as the vagina, uterus, and cervix is crucial for diagnosing, treatment planning, and monitoring of female reproductive health conditions. Similarly, the segmentation of newborn skeletons is vital for detecting developmental abnormalities, congenital anomalies, and growth disorders. Furthermore, it plays a critical role in building the maternal pelvic model, which includes a comprehensive fetal model with articulated joints, especially important for simulating childbirth.

Traditional segmentation methods often struggle with complex anatomical structures and variations in organ shapes and sizes. Deep learning approaches offer promising solutions by leveraging large-scale datasets and sophisticated neural network architectures to learn complex relationships and features directly from medical images. This facilitates more accurate and efficient segmentation.

A kind of CNN model commonly used for image segmentation is **U-Net**, which was initially proposed by Ronneberger et al. [38]. The U-Net architecture is characterized by its symmetrical U-shaped structure, hence the name U-Net. The network can be divided into two main parts: the contracting path and the expanding path. Each part serves a distinct purpose:

- Contracting Path: The contracting path employs a convolutional block and max pooling for downsampling. The convolutional block consists of convolutions, batch normalization, and an activation function. While max pooling reduces computational complexity and enhances noise robustness, it also has the drawback of information loss. After each downsampling step, the number of feature channels is doubled. Dropout is often employed to address overfitting, where a model becomes too specialized to the training dataset and fails to generalize well to unseen data. By randomly dropping units from the network during training, dropout simulates the use of multiple models for predictions, reducing overfitting.

- Expanding Path: The expanding path decoder consists of an up-convolution operation, concatenation with the corresponding feature map from the contracting path, and two convolutional blocks. This part reduces the size of the feature map using a convolutional layer, and then expands the feature map using up-convolution. Up-convolution involves zero-padding

the pixels before applying convolution to the padded image. The unique aspect of the U-Net structure is the symmetric connection between the contracting and expanding paths, allowing information to flow seamlessly between the two sides.

While U-Net is a powerful and widely used architecture for image segmentation, it has some drawbacks. U-Net requires a high amount of memory, especially when processing large input images. Despite its ability to capture local and global features through skip connections, U-Net may still struggle with capturing long-range dependencies and contextual information in large images or in cases where objects of interest are highly context-dependent. To get the best performance, U-Net demands considerable amounts of annotated training data.

Traditional deep learning models, such as convolutional neural networks and recurrent neural networks, work well with data that is represented by grid-like structures (such as images or sequences). However, they may not work well with data that has irregular or non-Euclidean structures, like graphs, point clouds, or meshes. **Geometric deep learning** [69], [70], a class of deep learning, addresses this limitation by designing neural network architectures and algorithms that can operate directly on unstructured data while preserving the inherent geometric properties. In the context of classical deep learning, dimensionality is closely linked to the number of features included in the data; meanwhile, in geometric deep learning, dimensionality pertains to the nature of the data rather than its feature count.

Geometric deep learning algorithms involve defining convolutional and pooling operations, as well as other layers, that are compatible with irregular structures. By leveraging techniques from differential geometry, graph theory, and topology, geometric deep learning enables the processing of complex data types commonly encountered in various domains.

In addition to conventional deep learning approaches, the use of **Generative Adversarial Networks** (GAN) [71] has also gained traction in medical image segmentation. GAN is a state-of-the-art method for deep learning generative modeling. The aim is to find patterns in input data, which allows the model to generate new instances that reasonably mirror the original data. We can deconstruct the term "generative adversarial networks" into three components as follows:

- Generative: this aspect involves learning a model that generates data, which explains the process of generating data in the context of a probabilistic framework.
- Adversarial: this term denotes that the generated outputs are pitted against the real data.
- Networks: this word indicates the use of artificial intelligence algorithms (i.e. machine learning, deep learning) to training tasks.

The structure of GAN consists of a generator model and a discriminator model as in Figure 5:

- Generator: is a crucial component of a GAN that generates novel and correct data from random noise. Typically depicted as a deep neural network, the generator learns the fundamental distribution of the training data through layers of adaptable parameters during training. By utilizing backpropagation, it adjusts its output to create samples closely resembling real data. The generator's effectiveness lies in its capability to produce diverse, high-quality samples that can deceive the discriminator.

- Discriminator: is for distinguishing between generated and real input. Acting as a binary classifier, the discriminator assesses input data to assign the likelihood of authenticity. Through continuous exposure to both real data and synthetic samples produced by the generator, the discriminator gradually improves its ability to discern between real and artificial data. This iterative process enables the discriminator to refine its parameters and enhance its discriminatory accuracy over time.

Training a GAN comprises an iterative process where both the generator and discriminator neural networks are simultaneously trained. Each of the two models has a unique loss function. The loss function of the generator evaluates its success in deceiving the discriminator, whereas the discriminator's loss function assesses its capability to differentiate between real and synthetic data. It is essential to highlight that training GANs can pose challenges and necessitate meticulous adjustment of hyperparameters, along with vigilant monitoring for problems like overfitting or instability.

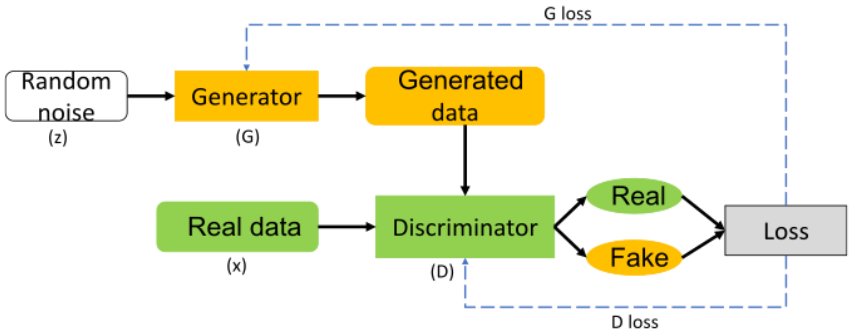


Figure 5. Overview of Generative adversarial networks (GAN) architecture.

Generative adversarial networks have developed into a variety of forms that are optimized for certain tasks or offer enhanced training stability and effectiveness. For example, Conditional GAN [72] expands upon the vanilla GAN for controlled creation by incorporating extra data like class labels to condition the generator and discriminator. By integrating deep CNN into the generator and discriminator, Deep convolutional GAN [73] aids in the production of high-resolution images with better stability. Another notable variant of GAN that has garnered attention is the Cycle-consistent GAN (CycleGAN) [74], developed specifically for unpaired image-to-image translation problems. Unlike vanilla GAN, CycleGAN learns to translate pictures from one category to another in a cycle-consistent way without the need for coupled training data.

2.1.2 Existing applications

Generative adversarial networks have been used extensively and successfully in numerous segmentation issues in recent years. For instance, Nie et al. [75] presented a residual adversarial network for distinguishing pelvic organs of the bladder, prostate, and rectum from MRI scans. The segmenter network employed convolution layers to produce the predicted mask, while a CNN served as the discriminator network for adversarial learning, aimed at refining the segmented organ structures (Figure 6).

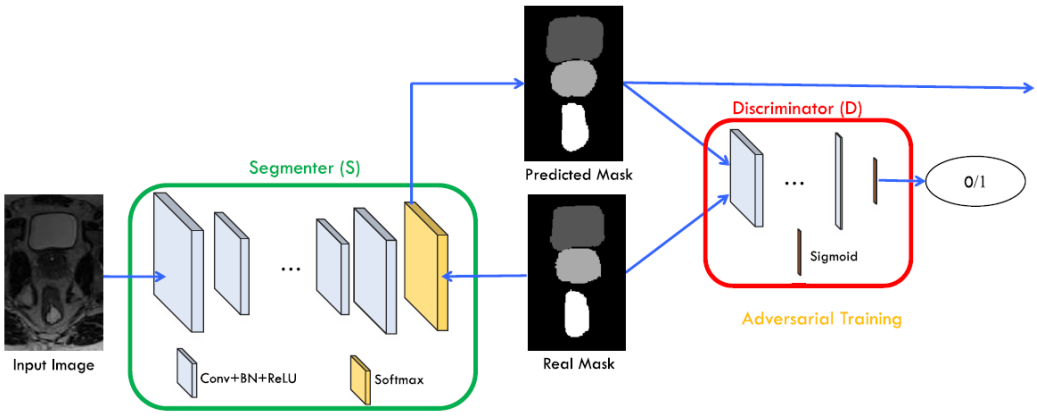


Figure 6. Architecture of segmentation networks using adversarial learning. A segmenter network to create the predicted mask and a discriminator network for adversarial learning [75].

Kalantar et al. [76] designed a deep learning model to synthesize pelvic MRI from CT scans. The study employed CycleGAN and U-Net++ to accomplish this task. Notably, the generated images enhanced sharpness throughout all pelvic slices, particularly in the bladder and bowel. In order to produce high-quality MRI scans from low ones, Li et al. [77] proposed a super-resolution model employing a GAN with cyclic loss and attention mechanism. The approach was applied to pelvic images obtained from healthy subjects.

Additionally, Tong et al. [78] based on GAN to develop a model for fully automated head and neck segmentation on CT and low-field MRI (Figure 7). The study employed a DenseNet, a deep supervised fully convolutional network, as the segmentation network. Following this, a discriminator network based on CNN was used to rectify anticipated errors and address discrepancies between the prediction and the ground truth at the image level.

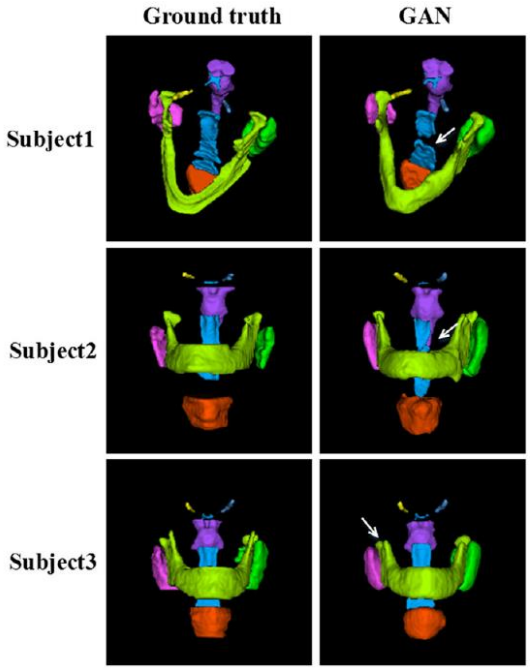
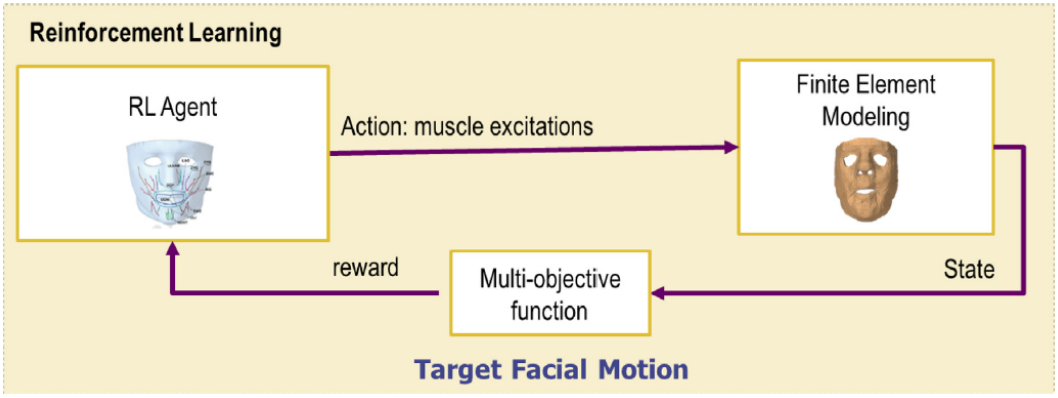
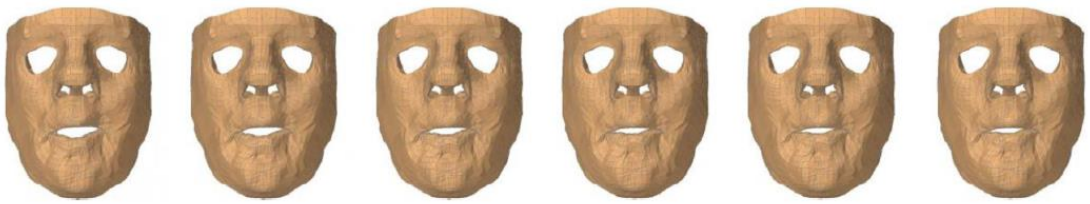


Figure 7. Head and neck segmentation using GAN compared to ground truth [78].

Facial analysis systems have experienced innovation because of deep learning algorithms, which make them able to accurately recognize facial features, expressions, and even motions. Nguyen et al. [79], for example, developed a model to predict the facial motion of patients with facial deformities. The integration of reinforcement learning with finite element modeling enhanced the model's capacity to facilitate personalized treatment planning and rehabilitation strategies (Figure 8). In another study, Nguyen et al. [80] applied a geometric deep learning method to recognize facial expressions from 3D point cloud data. This approach directly leveraged the geometric structure inherent in the data, leading to more precise recognition outcomes.



a)



b)

Figure 8. a) Summary of the innovative integration process linking reinforcement learning with finite element modeling, b) Facial animation for motion with symmetry [79].

Yolcu et al. [81] analyzed customers' facial expressions using convolutional neural network-based models. The proposed system began by detecting frontal face poses, then segmented key facial parts to generate iconized face images. Furthermore, Kim et al. [82] presented the application of U-Net for segmenting facial wrinkles, introducing a novel loss function designed to consider wrinkle thickness (Figure 9). The study utilized a semi-automatic labeling approach to enhance accuracy and efficiency in wrinkle segmentation.

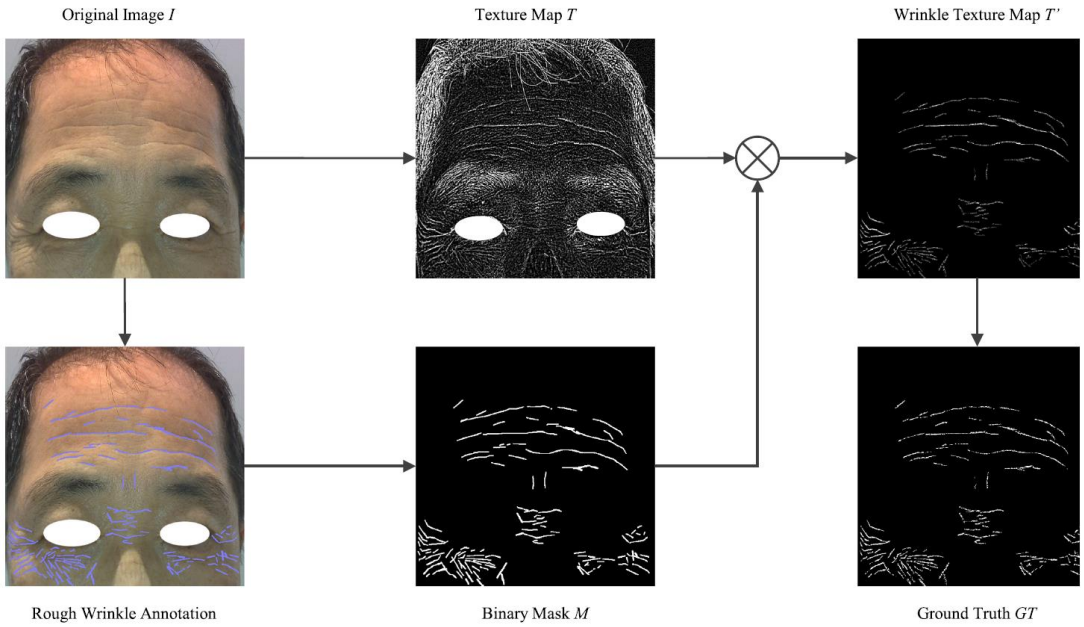


Figure 9. Overview of a wrinkle segmentation model. Wrinkle regions were labeled to build the binary mask M. The source picture I used to produce the suggested texture map T. After that, the wrinkle texture map T' was produced by multiplying the non-wrinkle text [82].

With the use of deep learning algorithms, bone segmentation has also advanced significantly. Belal et al. [83] introduced the segmentation of human bone into 49 distinct parts from the CT scans using convolutional neural networks (Figure 10). Schnider et al. [84] improved the bone segmentation through the utilization of segmentation networks that were trainable end-to-end, which amalgamated multiple 3D U-Nets. Cui et al. [85] designed a system based on deep learning to segment tooth and alveolar bone from cone-beam CT images. To segment teeth, an initial step involved a network that identified the foreground region of the upper and lower teeth. After that, a specialized deep network employed detailed geometric data to outline each tooth accurately. In the segmentation of alveolar bone, a tailored filter-enhanced network initially boosted intensity contrasts along bone borders.

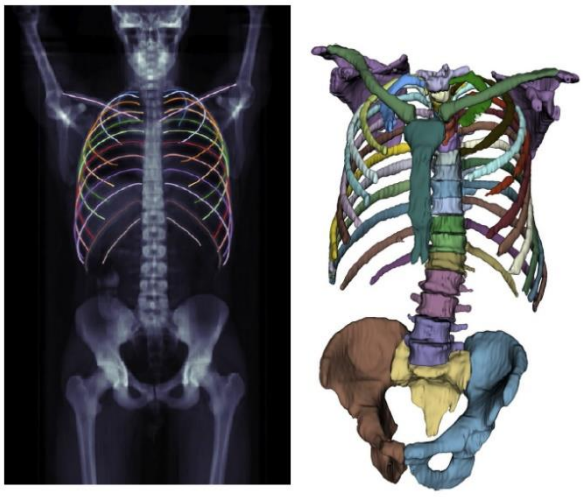


Figure 10. Segmentation and reconstruction of bony parts from CT scans [83].

In addition, Zeng et al. [86] integrated the attention method and deep supervision technique into the V-Net model to segment fetal head and skull structures from ultrasound images. Moreover, a multiscale loss function to comprehend supervision was presented. McCay et al. [87] employed fully convolutional neural networks to classify the movements of infants' bodies as in Figure 11. The study extracted normalized feature sets based on poses, histograms of joint orientation, and joint displacement to be utilized within deep learning frameworks.

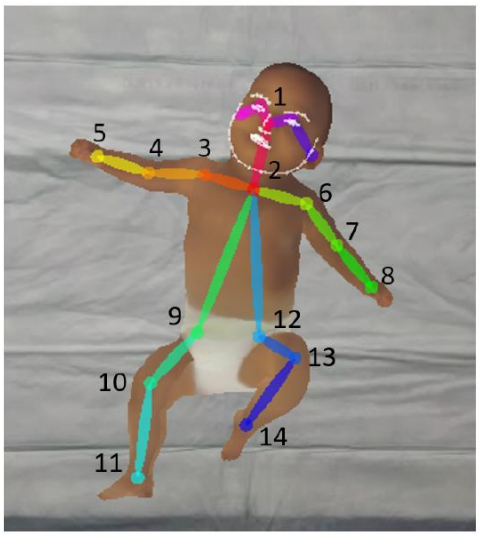


Figure 11. The skeleton output and its corresponding joint reference numbers [87].

2.1.3 Point cloud dataset

Point cloud data is a collection of points in a three-dimensional coordinate system. These points exhibit the external surface of an object or scene. Each point in the cloud has its own set of coordinates (x, y, z), representing its position in space, and may also include additional information such as color, intensity or reflectivity. Figure 12 illustrates a point cloud representation for the 3D rabbit [88]. Point clouds are commonly used in various fields such as architecture, engineering, urban planning, and computer graphics. They serve as a detailed representation of real-world objects or environments, allowing for accurate measurements, analysis, visualization, and modeling. Additionally, point cloud data is often processed and manipulated using specialized software to extract meaningful information, create 3D models or perform tasks such as object recognition and segmentation.

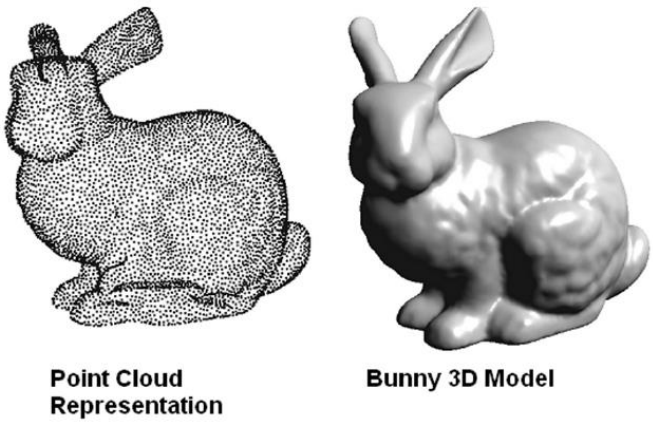


Figure 12. Point cloud depiction of a three-dimensional rabbit [88].

Point cloud classification and segmentation play vital roles in comprehending scenes, prompting numerous explorations into these areas. Machine learning approaches are the most commonly used methods for data analysis. In the field of point cloud classification and segmentation, deep learning is widely employed. Methods for point cloud segmentation can be categorized into three primary groups: projection-based methods, voxel-based methods, and point-based methods.

Projection-based methods focus on generating multi-viewpoint representations of point clouds, followed by learning the 2D features from these projections using 2D convolutional operators. The essence lies in constructing a robust understanding of the scene from various perspectives. Subsequently, the extracted features from the multi-viewpoint clouds are effectively fused. This fusion enhances the richness of information by integrating data from diverse viewpoints, culminating in a more comprehensive representation of the scene. Su et al. [89] introduced a CNN-based framework for recognizing 3D shapes. Initially, the framework identified shapes from individual views with high accuracy, surpassing current 3D shape descriptors. Then, it integrated information from multi-view into a single shape descriptor to enhance recognition (Figure 13). Wei et al. [90] presented two view-based graph convolutional network (GCN) models for 3D shape recognition, utilizing a graph representation of various perspectives. They established a view graph where each view served as a node and developed two GCNs to hierarchically learn discriminative shape descriptors, taking into account relationships among views.

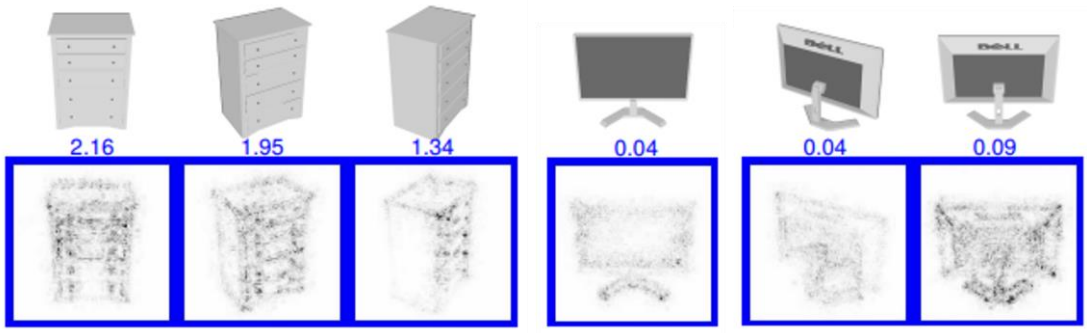


Figure 13. Top perspectives where objects were most salient [89].

Voxel-based methods involve partitioning the point cloud into a regular grid of volumetric elements and converting the point cloud into a voxel-based representation. This approach enables the application of convolutional neural networks, which are well-suited for regular grid structures. Voxel-based methods are useful when spatial relationships across the entire scene are important. Zhi et al. [91] suggested a volumetric CNN model called LightNet, aimed at tackling the challenges of real-time 3D shape identification through multiple learning tasks (Figure 14). The model concurrently predicted labels from both entire and partial objects. Wang et al. [92] also proposed a CNN model based on voxels to recognize 3D shapes, using both surface normal vectors and voxels of objects as inputs. To implement the convolution layers, the study introduced a module that captured identifiable characteristics for 3D visual analysis but substantially lowered the required parameters.

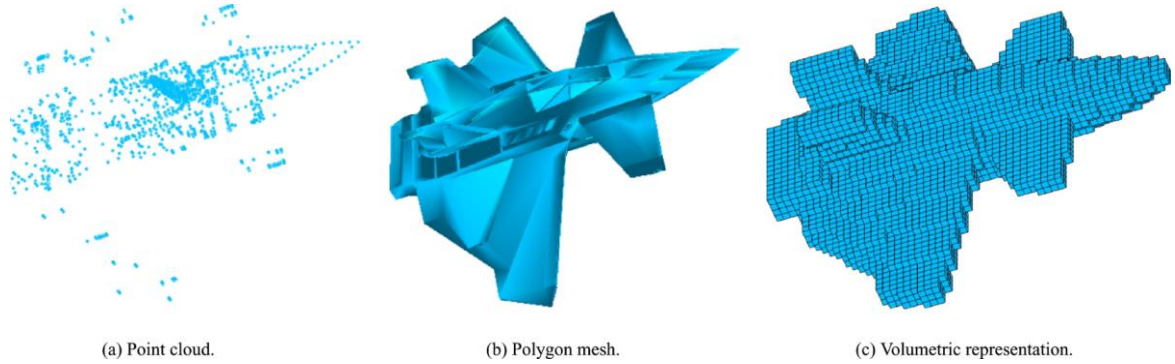


Figure 14. Converting the point clouds into structured 3D regular voxels [91].

The computational expense associated with the two aforementioned methods is significant, leading to the development of neural networks designed to operate directly on raw point clouds. In point-based methods, each point cloud is typically processed individually, which allows for the feature extractions and deep learning algorithms directly on the raw point data. Among these methods, PointNet [93] emerged as one of the earliest approaches to utilize deep learning for handling original point clouds (Figure 15). It revolutionized the field by capturing both local and global features without relying on intermediate representations such as voxels or meshes. PointNet streamlines the processing pipeline and facilitates more efficient analysis of point cloud data.

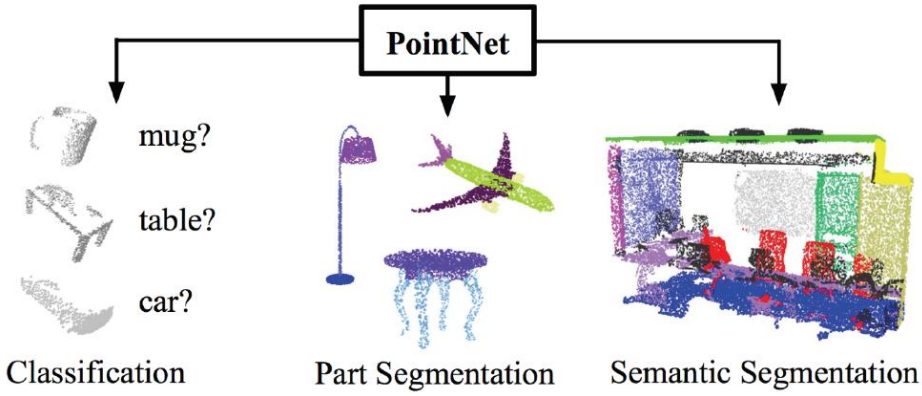


Figure 15. PointNet – one framework for several tasks [93].

2.2 Deep learning-driven prediction

2.2.1 Summary of used methods

In recent years, the applications of deep learning techniques in biomechanics have witnessed tremendous progress, completely changing our understanding and prediction of soft-tissue mechanical properties in the human body [94], [95], [96]. Among these, the prediction of deformation of pelvis soft tissue has enormous potential to improve diagnostic accuracy, treatment planning and patient outcomes in a variety of medical specialties such as obstetrics and gynecology. Soft-tissue displacements in the pelvis region can be influenced by a multitude of factors, including anatomical variations, pathological conditions, and external forces. Within the context of this thesis, we specifically investigate the deformation of pelvic soft tissue during childbirth. Accurate prediction of these displacements is crucial for tasks ranging from preoperative planning to real-time image-guided interventions. Traditionally, computational

models based on finite element analysis and biomechanical principles have been employed for such predictions. However, these models often rely on simplified assumptions and lack the capacity to capture complex nonlinear relationships inherent in soft tissue behavior.

2.2.1.1 Recurrent neural networks (RNNs)

The emergence of deep learning, particularly recurrent neural networks (RNNs), has opened up new avenues for more accurate and efficient soft-tissue displacement prediction. Recurrent neural networks are a category of neural networks enabling the utilization of prior outputs as inputs while maintaining hidden states. The primary function of these networks is to process sequential data, wherein the network's setup preceding the memory cycle influences subsequent output node information. This typical structure entails interconnected nodes within the hidden layers, where inputs to the hidden layer comprise not only those from the input layer but also outputs from preceding hidden layers.

The basic idea behind RNNs is that they contain loops that allow information to be retained over time. The network receives input x_t at time step t and computes the output $\hat{y}_t$. It also calculates the internal state update h_t. Subsequently, it transfers this internal state information from one network time step to the next. Because information is internally transferred within the network, we label these networks containing loops as recurrent networks. Figure 16 illustrates the unrolling form of the loop over time. In this representation, we can explicitly express the weighted matrices: first, the weights that transfer the input to the hidden state; then, the weights that transfer the hidden state to the output; and last, the weights that transfer the previous hidden state to the present hidden one. At every time step, we use the same weighted matrices. For RNNs, the loss is calculated at every time step, and the overall loss is obtained by adding the individual losses.

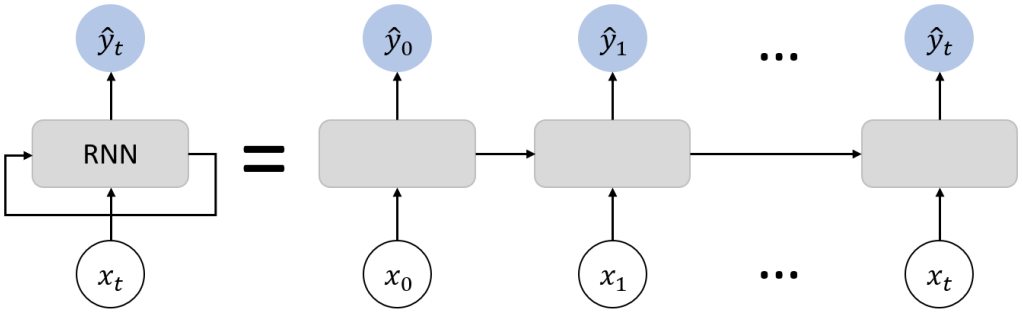


Figure 16. The expansion mode of RNN over time.

Besides their advantages, one significant issue with RNNs lies in the problem of vanishing and exploding gradients. The problems can be attributed to several factors. One way to conceptualize each RNN unit is as an individual neural network unit that is linked to the one after it. RNNs feature three states: an input state, an output state, and an internal memory state that holds onto data from earlier units. We advance through the network similarly to how we would in a feedforward network, predicting the output of each RNN unit and transmitting the data forward. Then, computing the loss of each unit to train the RNN, the total loss is obtained by summing up the losses of all units. After that, the network performs backpropagation, which leads to issues. This is due to the fact that backpropagation, particularly when working with lengthy sequences, requires computing many gradients to adjust the weights. As a result, by the time we backtrack to the first unit, multiple gradients have been computed. If gradients are less

than 1, the multiplication of many small numbers results in what is known as a vanishing gradient. If the gradients are more than 1, the multiplication of only a few numbers leads to what is referred to as an exploding gradient. The issue of exploding gradients can be alleviated through a technique known as "gradient clipping", which aims to limit excessively large gradients to prevent them from escalating. As for the vanishing gradient issue, it can be addressed by employing various strategies such as adjusting the activation function, optimizing weight techniques, and network structure.

A variant of the RNNs known as the **Long Short-term Memory** (LSTM) network, introduced by Hochreiter and Schmidhuber [35], was developed specifically to tackle the issue of vanishing gradients. In the LSTM architecture, there are two main components: the cell state and the hidden state. Throughout each iteration of the training process, the cell state is updated. This update process involves passing through three gates within the hidden state before the operation is repeated. Each gate plays a crucial role in determining the amount of information to retain and discard. These gates effectively manage gradients, ensuring they remain stable and preventing issues of vanishing and exploding gradients. The gates consist of standard neural network layers such as the *sigmoid* and pointwise multiplication. These three gates are referred to as: the *forget* gate, *input* gate, and *output* gate. The *forget* gate examines the information from the preceding cell state and determines which information needs to be retained and which can be disregarded before updating the cell state. This is made up of a sigmoid layer, with 0 representing forgetting, and 1 representing remembering. The *input* gate identifies which information is to be modified concerning the prior cell state. It uses a sigmoid layer to perform this evaluation, followed by transmitting these values via a tanh layer to generate new values. The *output* gate synthesizes the information from the previous stage to produce an updated state. Every time the training procedure is performed, the aforementioned loop is repeated, and the weights are updated at each stage. In terms of mathematics, the weight updates at each stage of the training process assisted in maintaining gradients that were constant or closer to 1, preventing RNNs' vanishing and exploding gradient issues.

A variant of the LSTM architecture is the **Bidirectional long short-term memory** (BiLSTM). The network was introduced by Schuster and Paliwal [97] to address the challenge of capturing long-term dependencies in sequential data. The distinguishing feature of BiLSTM lies in its ability to process sequences from both ends, unlike traditional LSTM that only moves forward through the data. This bidirectional processing enhances the network's understanding of context. Moreover, BiLSTM excels in learning long-term dependencies, making it well-suited for tasks involving sequential data analysis. By leveraging bidirectional processing, this model mitigates the risk of information loss that may occur in unidirectional LSTM setups, ensuring a more comprehensive representation of the input sequence. However, it is noted that BiLSTM has a slower processing speed and longer training time compared to LSTM. Consequently, we advise employing BiLSTM only when its unique bidirectional capabilities are essential for the task at hand.

2.2.1.2 Physics-informed neural networks (PINN)

Over the last decade, we have seen an incredible expansion in computer power, which has led to breakthrough developments in deep learning, particularly in deep neural networks. A unique deep neural network framework called physics-informed neural networks (PINN) has been created recently. PINN effectively combines the fundamental physical principles conveyed by partial differential equations (PDEs) with neural networks. By leveraging this inherent physical

understanding as a foundation, PINN can be trained as surrogate models for solving PDEs with minimal labeled data or sometimes no labels. Fully connected feedforward neural networks serve as the central component of the surrogate models to forecast outputs within the domain of differential equations. The differential operator in the governing equations is computed using automatic differentiation (AD) [98]. By traversing the backward chain in the neural networks, automatic differentiation – a continuous grid-free differentiation technique – can analytically compute derivatives.

In finite differences, the derivatives are approximated utilizing values of the original function assessed at specific sample points. Their basic definition relies on the limitation of a derivative. For instance, the derivative of a multivariate function f with respect to x is expressed as follows:

$$f'(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

in which ε is a small value. This approach offers the benefit of simplicity in execution, yet it comes with the drawback of having to conduct $O(n)$ evaluations of f to obtain a gradient in n dimensions. It is evident that computing $\frac{\partial f}{\partial x_i}$ entails assessing f twice: one for $f(x_1, x_2, \dots, x_i \dots, x_n)$ and one for $f(x_1, x_2, \dots x_i + \varepsilon, \dots, x_n)$. Moreover, it demands thoughtful selection of the ε number, and the outcome may lack precision when dealing with nonlinear functions.

Given that the neural networks embody composite functions, automatic differentiation iteratively calculates the derivatives by the chain rule. As a technical concept, AD denotes a particular set of methods that calculate derivatives by aggregating values throughout code execution to produce numerical derivative assessments rather than derivative expressions. There exist two versions of AD: *forward* mode and *reverse* mode. The forward mode is a combination of symbolic and numerical differentiation, although it produces numerically accurate results, it uses a lot of resources because it necessitates one traversal through the computational graph for each input parameter. In contrast, the *reverse* mode demands only two passes through the computational graph: a forward pass, which calculates all variable values, followed by a backward pass, which calculates derivatives. The reverse mode is widely adopted in practical applications, such as its utilization in TensorFlow.

Considering a function $y = f(g(h(x)))$, the inputs of the function are transmitted downwards in the forward pass as follows:

$$\begin{aligned} v_0 &= x \\ v_1 &= h(v_0) \\ v_2 &= g(v_1) \\ v_3 &= f(v_2) \\ y &= v_3 \end{aligned}$$

The derivatives are calculated in the backward pass using the chain rule. For example, the derivative $\frac{\partial y}{\partial x}$ is computed as follows:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial v_3} \cdot \frac{\partial v_3}{\partial v_2} \cdot \frac{\partial v_2}{\partial v_1} \cdot \frac{\partial v_1}{\partial v_0} \cdot \frac{\partial v_0}{\partial x}$$

Automatic differentiation proves significantly more productive than finite difference in scenarios with high dimensional inputs. AD can be used recursively n times for computing *n*-th order derivatives. Nevertheless, because of the potential for inefficiency and numerical instability associated with this stacked approach, other techniques such as Taylor-Mode automatic differentiation have been developed. It is important to highlight that AD allows the differentiation of neural networks, enabling the handling of noisy data.

A typical PINN approach is depicted in Figure 17 to solve differential equations. The input (x, t) of the neural network in the PINN framework corresponds to the training point coordinates, which can be divided into three parts. These are the samples $(x^i, 0)$ representing the initial condition, the samples (x^b, t^b) reflecting the boundary condition, and the collocation points (x^f, t^f) located within the domain of the equation. To compute the predicted value $\hat{u}(x_i, t_i, \theta)$ that corresponds to the input point (x_i, t_i). The parameters θ of the neural network (NN) include weights W and biases b. Automatic differential of the NN is employed to compute the partial derivatives of $\hat{u}(x_i, t_i, \theta)$. These derivatives are then submitted to the governing equations, and boundary/initial conditions to compute the errors of residuals for the aforementioned components. The loss function is defined as the linear sum of errors. Afterward, the predicted values are driven towards convergence with the exact values by minimizing total loss using an optimizer to adjust the parameters of the neural network.

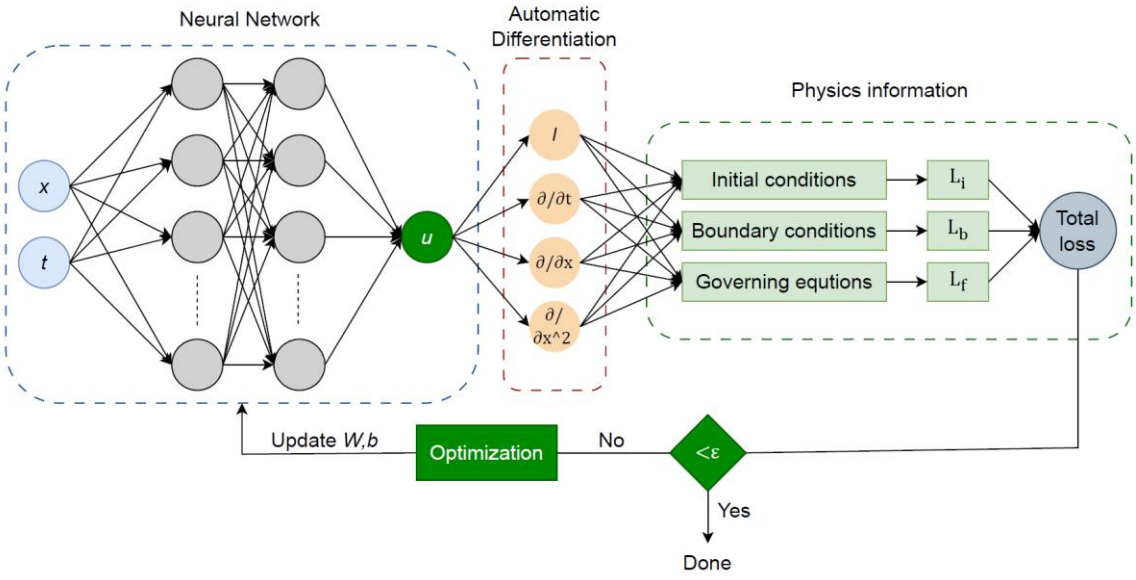


Figure 17. A schematic of Physics-informed neural network.

To elaborate on the concepts behind physics-informed neural networks and facilitate comprehension, we compare some points between PINN and traditional numerical methods as the following:

- Modeling approach:
  - PINN is a mesh-free method that directly learns the underlying physics of a system from data, and incorporates the governing equations and boundary/initial conditions into the loss function.

- Traditional numerical methods like finite difference, finite element, and finite volume methods discretize the domain into a mesh and solve the algebraic equations iteratively to approximate the solutions of partial differential equations.

- Computational cost:
  - During PINN training, neural network parameters are optimized using methods such as stochastic gradient descent and backpropagation. While training can require expensive computation, once trained, evaluation is often completed quickly.
  - Traditional methods involve solving large systems of equations iteratively, which can need a lot of processing power, especially for high-dimensional or finely discretized problems.

- Generalization:
  - PINN has the potential for better generalization to unseen data, as they learn the underlying physics from data and can capture complex patterns.
  - Traditional numerical methods are limited in the ability of generalization since the assumptions and approximations are made during discretization and solving procedures.

- Accuracy:
  - PINN substitutes the differential operation with automatic differentiation, bypassing the necessity for differential discretization and thereby mitigating discretization and truncation errors. This enables PINN to achieve solutions of high accuracy.
  - Traditional numerical methods highly depend on the chosen discretization scheme, mesh resolution and numerical stability of the solver, so their accuracy and robustness may struggle with high-dimensional nonlinear problems.

- Data dependency:
  - PINN requires data to train the neural networks, but they can also incorporate physics-based constraints to guide the learning process.
  - Traditional methods rely solely on the equations governing the physical system and do not directly utilize data for their computations.

2.2.1.3 Neural ordinary differential equations (NeuralODE)

Differential equations and neural networks represent two prominent modeling frameworks that are widely employed in scientific and technological domains. For centuries, differential equations have served as a fundamental tool for modeling a broad range of phenomena. Neural networks, on the other hand, have gained global prominence in recent decades to handle a variety of works like natural language processing and imaging identification. Recently, attention has shifted towards merging these methods into a blended approach known as neural differential equations.

We can conceptualize a neural network as a single large function composed of nested functions. However, determining the appropriate number of layers is not straightforward. Typically, fewer layers suffice for simpler problems or mappings, but as complexity increases, adding more layers is often beneficial. Yet, training a neural network with a large number of layers presents

challenges, primarily due to overfitting and vanishing gradients. A residual neural network (ResNet) is essentially identical to a standard neural network, with just one small modification. In a traditional neural network, the output of one layer is passed directly to the next layer. However, in a ResNet, we do not merely pass the output of one layer to the next; instead, we feed the output of the current layer plus the input of the same layer into the subsequent layer. A method to solve a first-order differential equation is Euler's Method, which operates in discrete time steps. Similarly, residual networks exhibit a formula, treating each layer as a discrete unit for weight optimization. Both approaches aim to approximate a function, but in the context of calculus, a differential equation solver like Euler's Method is used to derive the function. In Neural Ordinary Differential Equations (NeuralODE) [99], we apply ordinary differential equation solvers to solve a structure resembling a residual network, yet with a continuous unit instead of discrete ones (Figure 18). NeuralODE is well-suited for addressing challenges in modeling time series data. It provides memory efficiency, the capability to handle irregular data, and robust generalization to unseen data.

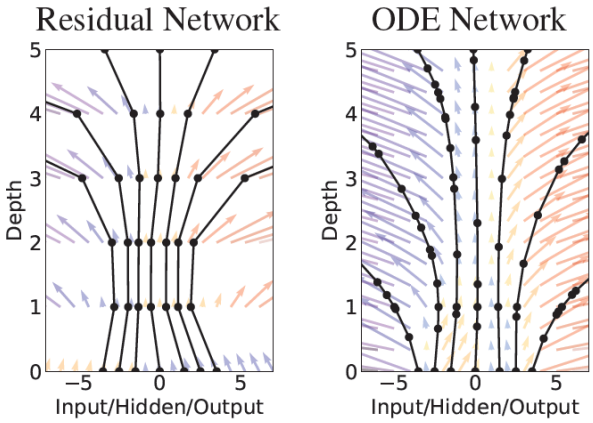


Figure 18. Difference between ResNet and NeuralODE. A residual network produces a discrete sequence; an ODE network produces a continuously transforming state [99].

2.2.2 Principal component analysis-based approach

When the number of attributes or dimensions in a dataset rises, the amount of data needed for meaningful outcomes grows dramatically, resulting in concerns like model overfitting, time-consuming computation, and poor precision of machine learning methods. With more dimensions, the potential attribute combinations also rise significantly, posing computational challenges for acquiring representative datasets. Consequently, tasks like clustering or classification are costly. Moreover, certain machine learning methods may be affected by the number of dimensions, necessitating more data to maintain accuracy degrees comparable to lower dimensional data. Thereby, feature selection and extraction techniques are employed to overcome this problem. The goal of dimensionality reduction methods is to minimize the input quantities while preserving the majority of the original data. In this section, we discuss the most widely used dimensionality reduction technique that is principal component analysis (PCA) [100].

The principal component analysis is a statistical method to translate a group of correlated variables into a group of independent variables. It is often utilized in data analysis and developing predictive models in machine learning. PCA is employed in data visualization to graphically represent complex data in 2D or 3D, simplifying interpretation. For feature

selection, PCA helps find crucial variables in data. It is also applied to data compression to downsize datasets while preserving key information. Additionally, PCA serves as an unsupervised learning approach to explore the connections between variables.

PCA determines a series of orthogonal axes known as principal components, which effectively encapsulate the most variance within the data. Principal components represent novel variables formed by combining or mixing the initial variables, which are arranged based on their declining significance. The majority of the variability in the data is captured by the primary principal component. Subsequently, each succeeding principal component captures the largest variance orthogonal to the previous one. PCA can be elucidated through a series of steps, unraveling the process of transforming a high-dimensional dataset into a lower-dimensional representation while retaining as much variance as possible.

The initial step is standardization to normalize the continuous original variables' scope to ensure each one makes an equal contribution to the analysis. This step is crucial because PCA is highly sensitive to variations in the original variables, which may bias outcomes if not addressed. Large differences in variable ranges can lead to a dominance of variables with wider ranges, thus standardization mitigates this issue by translating the data to equivalent scales, resulting in all variables being brought onto a uniform scale.

Next, the covariance matrix is computed to understand the interrelationships between the variables in the input dataset and to identify any duplicate information caused by strong correlations. Computing the eigenvectors and eigenvalues of the covariance matrix follows in step 3, determining the principal components of the data.

Step 4 involves generating a feature vector to decide whether to retain all principal components or dismiss those with lower relevance, which are characterized by low eigenvalues. We then form a matrix of vectors known as the feature vector, using the selected principal components. The resulting dataset will have a *p*-dimension if we retain *p* eigenvectors.

Finally, in step 5, the data is reoriented along the axes of the principal component. Except for the standardization step, no alterations are made to the data in the preceding stages; instead, principal components are selected and a feature vector is constructed. However, the input data is aligned with the original axes or initial variables. This step utilizes the feature vector generated from the eigenvectors to transform the data from its original axes to those defined by the principal components.

2.2.3 Existing applications

Various applications of RNN models in biomechanics demonstrate their efficacy in predicting complex physiological phenomena. For example, Burton et al. [101] employed BiLSTM models to forecast joint contact and muscle forces based on predictor attributes obtained from gait lab data (Figure 19). These predictor attributes comprised various factors such as stature, mass, and angles reflecting pelvic orientation, knee flexion, ankle plantar, and dorsi flexion. Despite eliminating reliance on specific subsets of predictor attributes, RNNs accurately anticipated patient mechanics. Said et al. [102] proposed the use of BiLSTM alongside an attention layer for predicting breast tissue deformation during mammography. Two phases were involved in applying the deformation: simulation of compression and estimate of the unloaded state. Numerous clinical datasets and realistic compression ratios were used to evaluate the approach. Furthermore, Zhang et al. [103] introduced a deep learning model of LSTM

combined convolutional neural networks to predict brain tumor growth. The method integrated 3D spatial and temporal images along with clinical data to simultaneously capture intra-slice spatial patterns, inter-slice relationships within 3D contexts, and temporal changes within sequences over time. Similarly, Lombardo et al. [104] developed LSTM-based models to anticipate future tumor locations in two motion directions and to dynamically adjust an existing contour, giving rise to the future tumor contour. The performance of these models was assessed using various metrics to gauge their precision in deformable multi-leaf collimator tracking, utilizing MRI data collected during patient treatments.

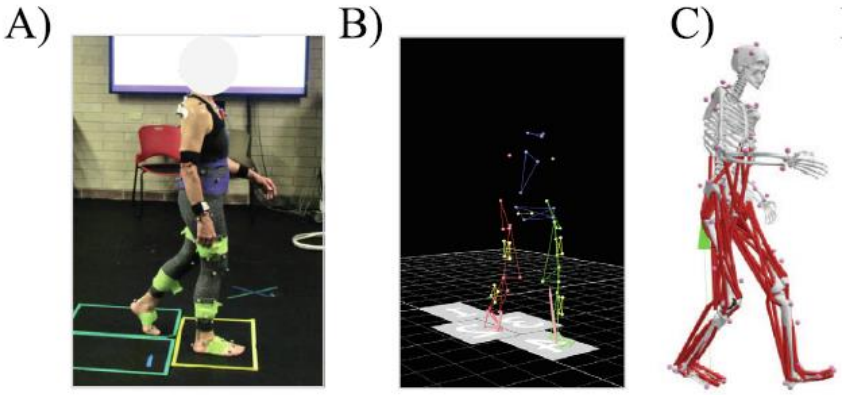


Figure 19. Workflow of estimating forces during activities. A) gather ground response force inputs and marker data. B) determine joint angles in musculoskeletal modeling from marker data. C) estimate mechanical properties from musculoskeletal modeling [101].

Physics-informed neural networks have been effectively applied across numerous domains including mechanics [105], [106], [107], material science [108], [109], and medical diagnostics [110], [111]. For example, Buoso et al. [112] suggested the use of physics-informed neural networks capable of tailoring to individual patients, offering rapid and dependable assessments of left ventricular mechanics as in Figure 20. This network efficiently created a function cardiac model from anatomical clinical images, requiring significantly less computational time compared to conventional finite element methods.

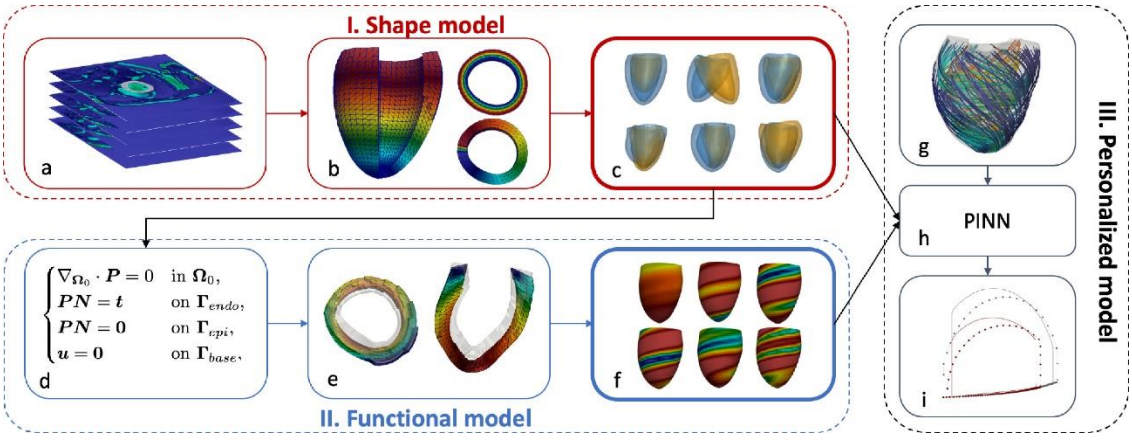


Figure 20. Outline the systematic steps for forecasting left ventricular mechanics [112].

Moreover, Li and Lee [113] provided the foundation for parameter identification and boundary force prediction using a PINN incorporating a boundary condition integrated estimate function. The efficacy of the proposed model was showcased through its application in determining

critical factors that define the ankle joint's dynamics. Zhang et al. [114] introduced a musculoskeletal framework based on PINN to predict joint kinematics and muscle forces from surface electromyograms. Datasets of walking trials and wrist motions were used to verify the model. The predicted outcomes showed the efficiency of the suggested model, which addressed the limitations of data-driven and physics-based methods. Additionally, Movahhedi et al. [115] presented a blended PINN designed to predict 3D flow-induced tissue dynamics and physical properties from 2D images. This approach combined an LSTM model representing soft tissue with a differentiable fluid solver. By utilizing pre-existing knowledge in solid mechanics, the algorithm translated the governing equation onto a discrete eigenspace.

In the applications of neural ordinary differential equations technique, Goyal et al. [116] integrated deep neural networks with NeuralODE for learning dynamical models from noisy and irregularly sampled data. The goal was to train two networks: one to implicitly represent measurement data and another to approximate the vector field, connected through integral forms of ODEs. Notably, this approach did not demand explicit noise estimation and accommodates varying temporal grids for dependent variables. Kashyap et al. [117] employed NeuralODE to examine the trajectories of brain network models, which replicate the brain's activity. These models were simulated over short trajectories using estimated initial conditions derived from observed functional MRI data. Yang et al. [118] implemented NeuralODE, where deep feature extraction was governed by an ODE parameterized by a neural network, to enhance deep learning explainability in glioma segmentation from multi-parametric MRI data. The schema depicted as in Figure 21. Visualizing the dynamics of MR images after interacting with the deep neural network and segmentation formation was achieved by solving the ODE.

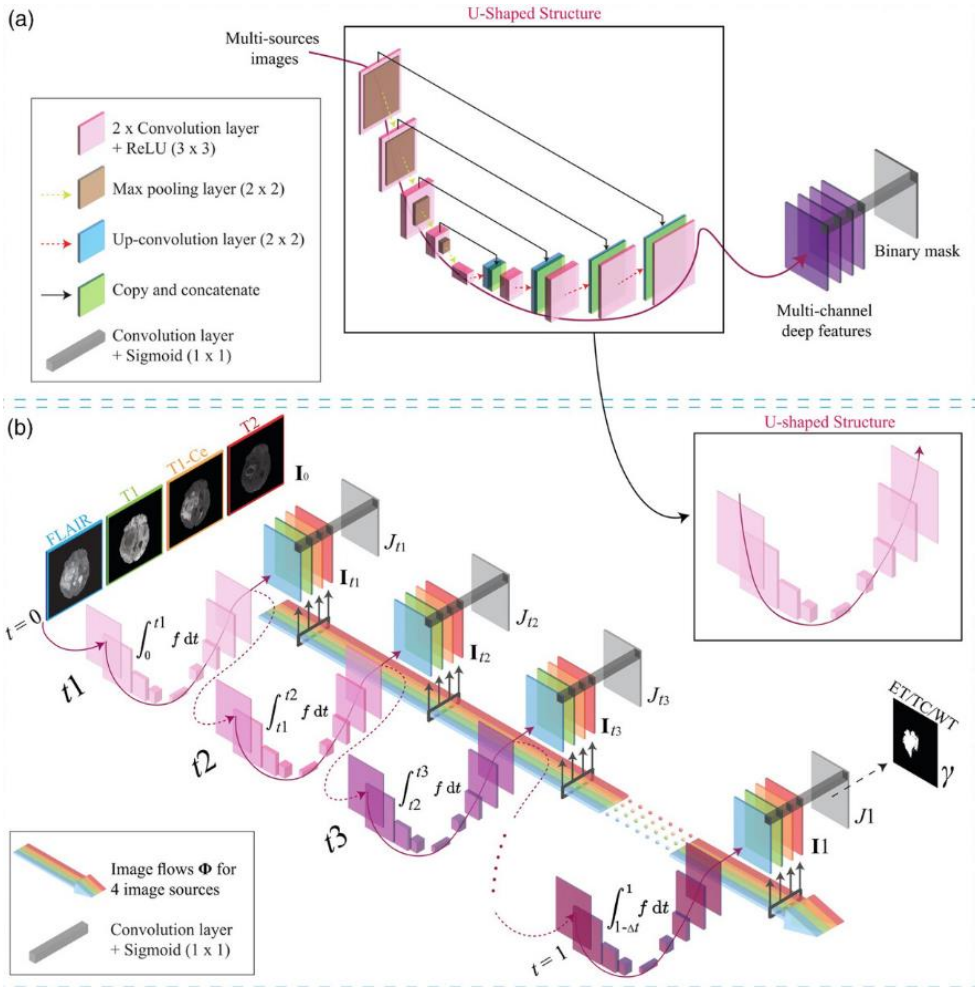


Figure 21. Structure of a Neural ODE model based on U-Net architecture for glioma segmentation [118].

Principal component analysis, a widely recognized statistical method, has been utilized to create models tailored to individual subjects. Samei et al. [119] constructed prototype models by applying a PCA model to the motion vectors of each patient. The ultimate model comprised a weighted fusion of predictions derived from all individual sub-models. In a subsequent study, Tanner et al. [120] tracked liver vessel landmarks and temporally extended them to achieve spatiotemporal forecasting based on the PCA model. Garau et al. [121] introduced a motion model that relied on regions of interest (ROI) identified from 4D computed tomography and 2D cine-MRI scans. The objective was to effectively address alterations occurring during the treatment of lung tumors. The approach employed PCA to construct the model based on ROI associated with anatomical structures.

2.3 Challenges and objectives of the thesis

Constructing a robust model for simulating the childbirth process necessitates the comprehensive construction of both the maternal pelvic system and the fetal body, complete with full joint articulations. While numerous models have emerged recently, they still have various limitations. One common shortfall is the omission of a fully articulated fetal body representation, with a predominant focus solely on the head and neck of the fetus. Additionally, the conventional numerical methods employed for solving the motion of soft tissue prove to be

time-consuming, hindering real-time simulation. Furthermore, there is a notable absence of the utilization of state-of-the-art deep learning methods to accurately predict pelvic soft tissue deformation in real-time scenarios. Overcoming these challenges is imperative for the development of a comprehensive childbirth simulation model that can accurately depict the complex dynamics of labor and delivery.

The objectives of this thesis are to develop and evaluate models for fetal representation and real-time soft tissue deformation. The main works are distributed into three chapters.

In Chapter 3, we develop and assess a generative adversarial model for the automatic and rapid segmentation of bone components within the entire neonatal skeleton (Figure 22). A dataset comprising CT images of newborns was gathered and segmented, with each 3D reconstructed skeleton segmented into separate bony segments. We propose a GAN model based on PointNet to conduct automated segmentation directly on the 3D point clouds. A comparative analysis with a pointwise convolutional neural network to showcase the accuracy and efficiency of our method will be presented.

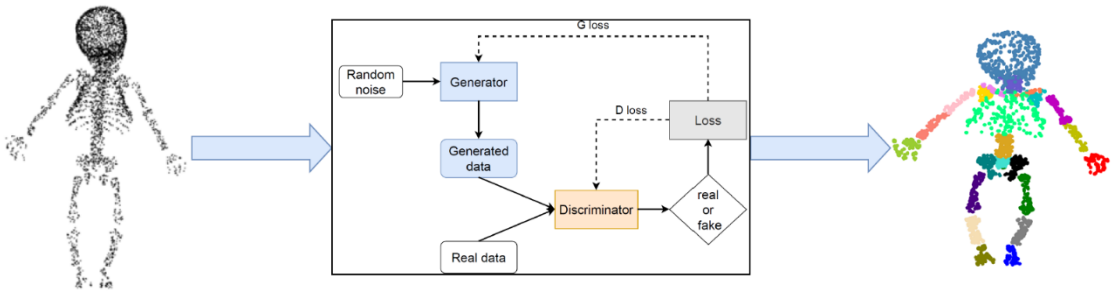


Figure 22. Newborn 3D skeleton part segmentation using GAN.

The point cloud data used in our study for newborn skeleton segmentation is derived from a CT database containing 124 subjects. Each subject undergoes manual segmentation of the whole skeleton using the software. Following this, we partition the skeleton into 23 sections. To enhance the data quality for training purposes, we augment it by applying rotation and resampling algorithms on the original points. As a result of this augmentation, the dataset comprises 3808 samples.

In Chapter 4, we present a novel method utilizing deep learning techniques to forecast the deformation of pelvic soft tissues (Figure 23). Specifically, we employ the Long Short-term Memory (LSTM) alongside a deep neural network (DNN) to address high-frequency oscillation signals. We also implement various learning approaches, including those with and without data dimension reduction. We construct a simulation-driven database utilizing the Hyperelastic Mass-Spring Model (HyperMSM) model to facilitate training and testing.

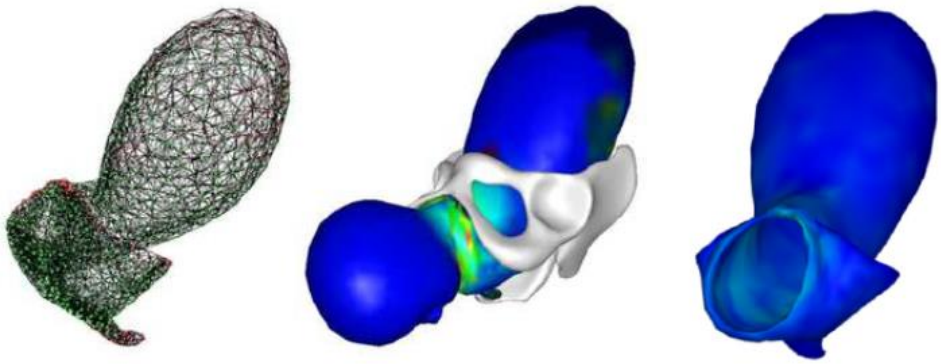


Figure 23. The prediction of pelvic soft-tissue displacement.

We generate the learning database by simulating childbirth processes. Our training method involves taking raw time-series data, including vertex positions, velocities, and interactive forces applied to nodes, and predicting nodal displacements. We choose to use our proprietary Hyperelastic Mass-Spring Model (HyperMSM) simulator for these simulations and to compile the database, as it has proven capable of simulating the biomechanical properties of soft tissues with a satisfactory level of accuracy. The model includes a fetal body, pelvic soft tissues such as floor muscles, vagina and uterus, as well as the bony structure of the pelvis. While simplified and not fully representative of the anatomical complexities of the female pelvis system, it is considered adequate for the immediate objectives of this study.

Using HyperMSM, we execute many simulations of the birthing process while keeping the same material characteristics. For these simulations, the time-step is fixed at 2 milliseconds in each case. While some other methods allow for variable time-steps, HyperMSM uses the projective dynamic technique for time integration, which demands a constant time-step throughout the simulation. In dynamic simulations, the time-step value setting is crucial: while a bigger time-step accelerates the simulation process, a lower time-step enhances simulation stability. We produce a database with entries made up of the vectors x_t, v_t, F and u_{t+h}. the deep learning models will use the first three vectors as input, and u_{t+h} as output. We simply consider unconstrained surface points in three dimensions. The model consists of 1653 surface nodes, resulting in vectors of size 3×1653. During each simulation, random external forces are applied to the surface of the pelvic floor muscles to simulate potential clinical intervention forces. We obtain a training database of 15396 frames concerning 8 simulations, and a test database of 18633 frames with respect to 6 simulations.

Extending the work in Chapter 4, **in Chapter 5**, we develop an appropriate predictive model by combining Physics-informed neural networks (PINN) and Neural ordinary differential equations (NeuralODE) within the mass-spring system (MSS) modeling framework to simulate real-time deformations of soft tissue subjects (Figure 24). We evaluate our innovative method on 2D and 3D mass-spring systems and a 3D cantilever beam. Lastly, we simulate and assess the deformation of a 3D uterus model under the influence of gravity loading.

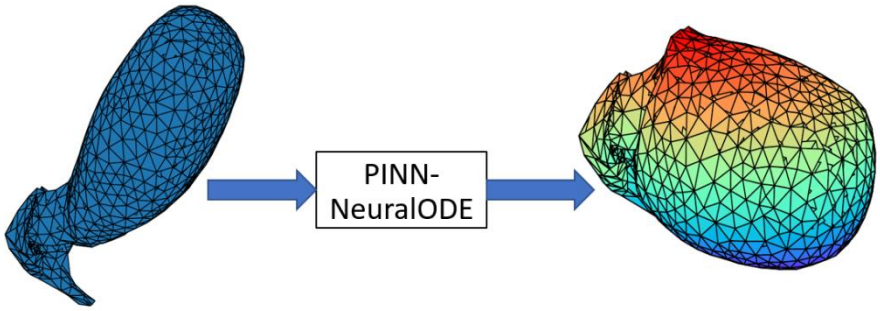


Figure 24. Physics-informed neural networks for predicting deformation of pelvic soft-tissue in real-time.

Chapter 3

Generative Adversarial Network for Newborn 3D Skeleton Part Segmentation

Research on childbirth simulations aims to anticipate and mitigate challenging delivery complications. Key to this endeavor is the reconstruction of the maternal pelvic model, which incorporates a detailed fetal model complete with articulated joints. Yet, the traditional methods of segmenting the diverse bone structures using classical image processing techniques are arduous and time-intensive. Thus, this chapter endeavors to introduce and assess a generative adversarial network (GAN) tailored to swiftly and automatically segment the bone components of the entire neonatal skeleton, offering a promising solution to this persistent challenge.

3.1 Introduction

In many computer-aided medical applications, the precise segmentation of every single bone in the individual body is significantly essential. Bone segmentation holds the potential to enhance the diagnosis of bone diseases, evaluates post-treatment progress, as well as guides imaging for treatment strategies such as surgical operation and radiotherapy. The consistent forms, locations and alignments of the skeleton structure make the segmentation task challenging and difficult. In particular, bony structures of fetuses affect to the childbirth process and relative to injuries of infants and mothers. Clinical experts have been concentrating their studies on understanding the process of birthing for a long time. In fact, extracted knowledge can be used to determine the causes of injuries to babies and mothers in labor works and investigate treatment strategies for complex deliveries. To achieve this objective, many methodologies have been used. One common method is to analyze historical observations to obtain correlation factors between a delivery situation and particular maternal–fetal variables. For example, neonatal brachial plexus palsy (NBPP) has been studied using a retrospective cohort [122]. Shoulder dystocia was identified as a potential risk factor, which is statistically relevant to the incidence of NBPP, but this problem cannot be forecasted with only shoulder dystocia symptoms. This approach fails to explain the mechanism and cause of injuries of the mother and fetus in the childbirth process.

Another method relates to the use of maternal and neonatal birthing simulators. Medical students and clinicians practice realistic normal and abnormal labor works with such simulators. For example, Macedonia et al. [123] presented a female robotic childbirth simulator, named Noelle, to train clinicians in operating vaginal delivery and improving skills in vaginal breech labor work. Dupuis et al. [124] evaluated transvaginal fetal head station by using a birth simulator. However, physical birth simulators have drawbacks when providing the physiological causes of birth-related damage processes. They are often designed for a typical delivery scenario with typical sizes of the mother and fetal anatomies. Thus, it is difficult to present patient-specific parameters. Moreover, physical simulators are restricted in their ability to describe mechanical properties exhibited commonly in the delivery such as deformation of soft tissue (e.g. uterus, pelvic organs). It is also impossible to adequately reproduce complex situations for example the uterus contraction.

In addition, childbirth computational modeling can be identified as a promising approach to study mechanisms of the delivery and injuries happening to the mother and fetus. Parente et al. [125] simulated the fetal head motion during the birth canal to investigate its impact on pelvic muscles using the finite element method. The 3D rotation of fetal shoulders was modeled to evaluate how shoulder dystocia damage to brachial plexus tension and birth pressure by Chen et al. [126]. Typically, fetal models primarily include the head. Only few studies integrate a detailed fetal body with joints [127], [128]. It is important to note that the articulated fetal body is of great important to accurately model newborn skeleton movement and associated delivery mechanics. In fact, fetal skeleton part segmentation contributes significantly to the development of a fetal model with full joint definition to understand better the childbirth process.

Recently, there has been a growing interest in point cloud learning, driven by its diverse range of applications in computer vision, automatic control and robotic fields. While deep learning has been highly effective in solving 2D vision tasks as well as has dominated the AI landscape, its application to point cloud learning is relatively new and faces distinct challenges. In particular, there is a burgeoning interest and development in deep learning methods for point cloud learning. In this study, we propose to use deep learning methods applied directly on 3D point clouds for the newborn skeleton part segmentation.

Deep learning methods have obtained great achievements in several domains such as computational biomechanics [129], [130], [131], computer vision [132], [133], medical diagnosis and healthcare [25], [134], [135]. Liu et al. [136] presented the deep multi-class network-based U-Net to segment pelvic bones of the lumbar spine, sacrum, left hip and right hip from CT imaging dataset. Other parts of the pelvic bone such as sacrococcyx, femoral head, femoral neck, ischium and pubis, were also segmented from MRI scan data by applying 3D U-Net in the study performed by Liu et al. [137]. Moreover, Zhang and Wang [138] developed a framework using PointNet++ architecture for the cervical vertebrae segmentation.

In this study, we develop the generative adversarial network (GAN) based on the pointwise multi-layer perceptron method to segment multiple parts of the fetal skeleton from 3D point cloud dataset. GAN is a state-of-the-art method of generative models, which was first introduced by Goodfellow et al. [71]. The structure of GAN includes two neural network models that are trained simultaneously. The first network is known as the discriminator with a duty to distinguish authentic and fraudulent samples. The other one is known as the generator to generate samples to trick the discriminator as actual samples with high likelihood. In recent years, GAN has been widely employed in many problems and achieved considerable success. For instance, Odena et al. [139] demonstrated a well-performed GAN for the image synthesis on the ImageNet dataset. An image-to-image translation was also solved by Zhu et al. [140] with the GAN and cycle-consistent loss algorithm that is able to learn to capture distinctive properties of one picture set and figure out how these qualities may be transferred into the other set. Moreover, Ma et al. [141] trained an unsupervised generative adversarial autoencoding network (UGGAAN) for the semantic segmentation of the objects in a real scene.

It is well-known that when the inherent inconsistencies exist in the data, conventional deep learning techniques designed for 2D images are not readily transferable to 3D point clouds. PointNet [93] stands out as an innovative approach that accepts point clouds directly as inputs and accomplishes permutation invariance through the use of a symmetric function. PointNet separately acquires point-specific characteristics using multi-layer perceptron and captures global characteristics through a max pooling layer. In the present work, we study the GAN

model based on PointNet algorithm to segment newborn skeleton parts directly from 3D point clouds. We also compare our approach with the pointwise convolutional neural network method [142].

3.2 Material and Methods

3.2.1 Newborn skeleton database and segmentation workflow

In this study, we collected post-mortem newborn CT scans from the New Mexico Decedent Image Database (NMDID) [143]. The CT scan database of 124 subjects aged from 0 to 10 months was obtained and processed. The average body length was 51.7 ± 11.5 cm and the average body mass was 4.6 ± 3.2 kg. The format of images was saved in DICOM format with sagittal, coronal and frontal axes.

Each subject of the full 3D skeleton structure was segmented through a threshold-based approach and the 3D Slicer software. Then, 23 bony parts were manually isolated and refined for each subject. Figure 25 describes the labelling scheme of the 23 bony segments (i.e. the skull, left- and right- (L- R-) clavicula, L- R- pelvis, L- R- femur, L- R- foot, L- R- hand, L- R- humerus, L- R- radius + ulna, L-R- scapula, L-R- tibia + fibula, vertebrae_C, vertebrae_L, vertebrae_S + coccyx, and vertebrae_T). A total of 119 subjects were retained and saved in Object File Format (OFF) format for further processing. Thus, 5 subjects were removed from the database due to missing or ambiguously identifiable segments in their scans.

We augmented the database by rotating and applying the resampling algorithms on the original point clouds to improve the quality of data for training. The obtained database after data augmentation includes 3808 instances. Thus, each subject dataset was transformed to have additional 31 datasets. Each subject and its data augmentation were orderly placed in the database. To split our database, we took the first 83 subjects and their data augmentation (about 70%) to the training set. Then, the next 12 patients and their data augmentation (about 10%) were used for the test set while the remaining 24 patients and their data augmentation (about 20%) were assigned to the validation set. This splitting process ensures that different subjects are stayed in different sets. Each subject had 2048 3D points clouds. Each point cloud belongs to one of 23 labels corresponding to 23 bony parts. All labels were represented in one-hot encoding format. Furthermore, the 3D data point clouds of instances were normalized in the range [-1,1] for each spatial dimension. We proposed the GAN-based PointNet method and pointwise convolutional neural network for the skeleton part segmentation. The segmentation workflow of the 23 bony parts of the newborn skeleton using our proposed deep learning methods from the 3D point clouds is shown in Figure 26. Note that the implementation of the proposed methods was performed by using the Google Colab engine.

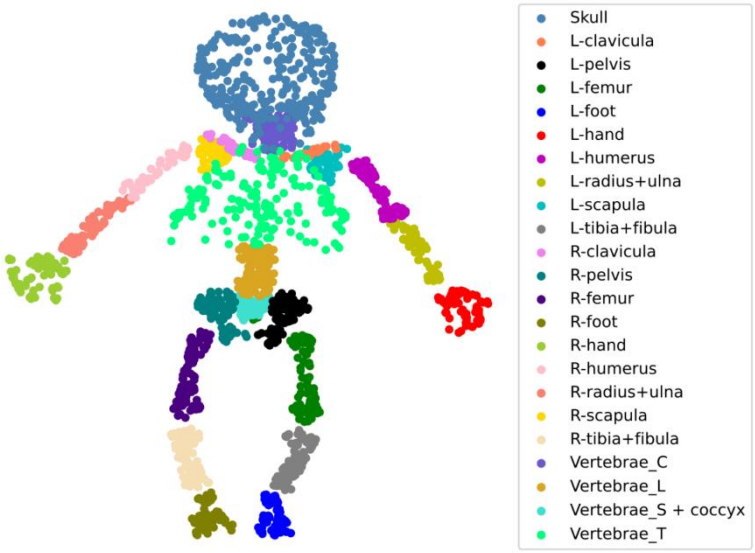


Figure 25. Illustration of the full 3D skeleton with associated description of the labels of 23 bony segments.

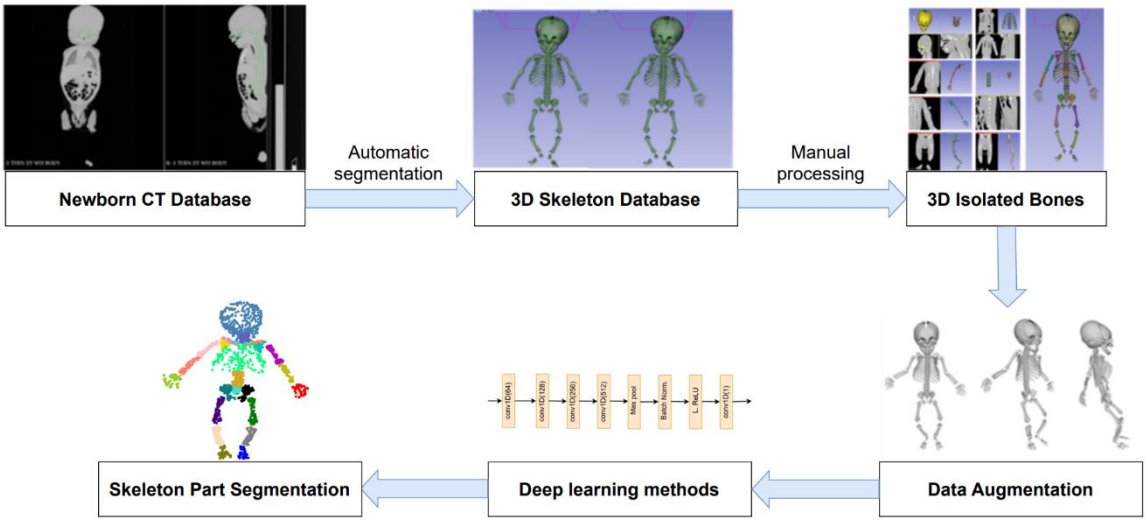


Figure 26. Implementation flowchart of the part segmentation of the newborn skeleton using deep learning methods from 3D point clouds.

3.2.2 Generative Adversarial Network (GAN)

The generative adversarial network is a cutting-edge generative model that is capable of understanding the pattern of training data and generating novel samples based on it. The fundamental finding of the GAN is the training procedure as a competition between two neural networks: a generator (G) to create new plausible samples and a discriminator (D) to differentiate which samples are fake and which ones are real. These networks are trained simultaneously. The training dataset for the discriminator is gathered from two origins that are ground truth used as positive samples during training, and fake dataset, which is generated by the generator, used as negative samples. While the discriminator undergoes training, the generator remains static, maintaining fixed weights as it generates samples for the discriminator's training. The discriminator distinguishes between true and fake data, and its weights are updated by the backpropagation according to the discriminator loss. The generator learns to generate counterfeit data by integrating the discriminator feedback that tries to trick

the discriminator into classifying the generator's output as authentic. The overview of a GAN is illustrated in Figure 27.

Generative adversarial networks are evaluated by the minimax loss function. A GAN can have two loss functions: one for generator and one for discriminator. The loss function can be mathematically expressed as follows:

$$\mathcal{L}(D,G) = E_x\left[\log\left(D(x)\right)\right] - E_z\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (21)$$

where $D(x)$ denotes the estimation of the discriminator about the likelihood of real data x when it is real, E_x is the predicted value across all real data. Input random noise is denoted as z, the output of the generator is $G(z)$, $D(G(z))$ denotes the estimation of the discriminator about the likelihood of a fake sample when it is real. E_z is the predicted value across all inputs according to the generator. The generator attempts to minimize the loss value while the discriminator aims to maximize it.

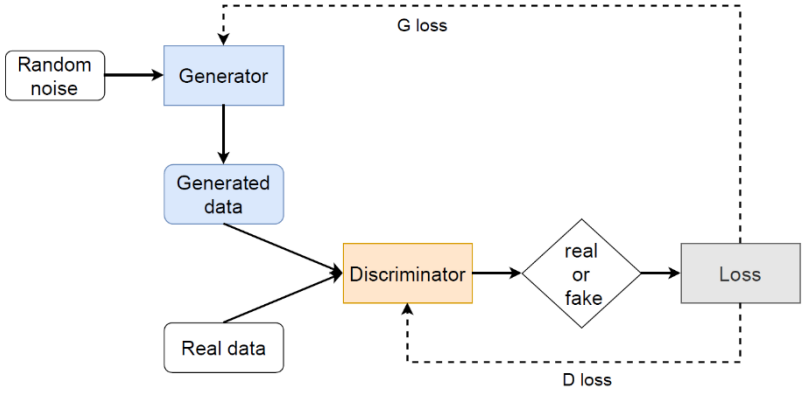


Figure 27. A schematic representation of a generative adversarial network architecture.

To build a generator network, we base on the PointNet model that is a novel neural network architecture to handle point clouds data proposed by Qi et al. [93]. PointNet reads unsorted point clouds straight away and ensures the permuting consistency of the points. The fundamental advantage of the PointNet is its resistance to input perturbation and manipulation. In addition, the model can be trained to describe a form using a sparse collection of key points. The main idea of PointNet is to use a family of symmetric functions by neural networks on altered points for estimating the generic function given on a set of points. Multi-layer perceptron (MLP) and max pooling functions are commonly used.

In order to perform the segmentation skeleton parts, the input of the generator network is a set of N point clouds $\{p_1, p_2, \ldots, p_N\}$ with each p_1 having a coordinate (x, y, z); the output is $N \times m$ scores, where m is the number of segmented parts (m = 23 in this study). The model includes spatial transformer networks, which aim to standardize the data using solid or affine transformations to ensure that each individual point undergoes independent transformation. A transformation block consists of many neural network layers so-called as T-net and an identity matrix. The T-net has convolutional neural networks 1D (conv1D) of features 64, 128, 1024 with batch normalization layer and the ReLU activation function, a max pooling layer, shared MLP layers with features of 512, 256 and a fully connected layer to reduce the size of features into a transformation matrix. The output of the T-net is added to the identity matrix. In the proposed generator model, the input point clouds $N \times 3$ are transferred to the transformation

block with output feature of 3. The segmentation network applies shared MLP to map N point clouds from 3 channels to 64 channels. After that, the second transformation block, which has the same structure to the first one, but the output includes 64 features, is employed. Next layers are shared MLP features of 128, 1024. Following that, we apply max pooling function to obtain a global feature in 1024-dimension, which is concatenated with the output of the second transformation block. Finally, a series of MLP features of 512, 256, 128 and m are used to achieve $N \times m$ output scores. Note that the last layer uses the softmax activation function. The Figure 28 shows the architecture of this generator network.

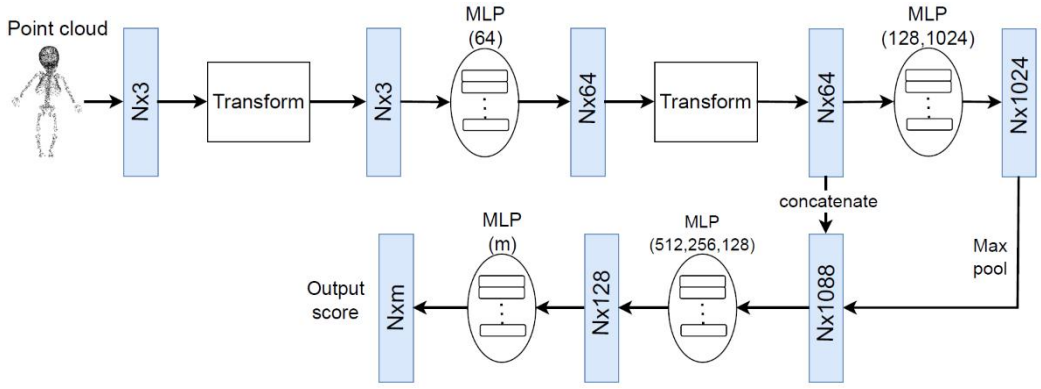


Figure 28. The structure of the generator model based on PointNet architecture.

For the discriminator network, we apply the convolutional neural networks (CNN). The input includes two sets (i.e. point clouds and associated ground truth part segmentation), point clouds and generated data from the generator model. We feed input data through conv1D features of 64, 128, 256 and 512 respectively. A max pooling layer, a batch normalization layer and Leaky ReLU activation function are provided. Finally, a fully connected layer is used for the likelihood score output. The flowchart of the discriminator model is described in Figure 29.

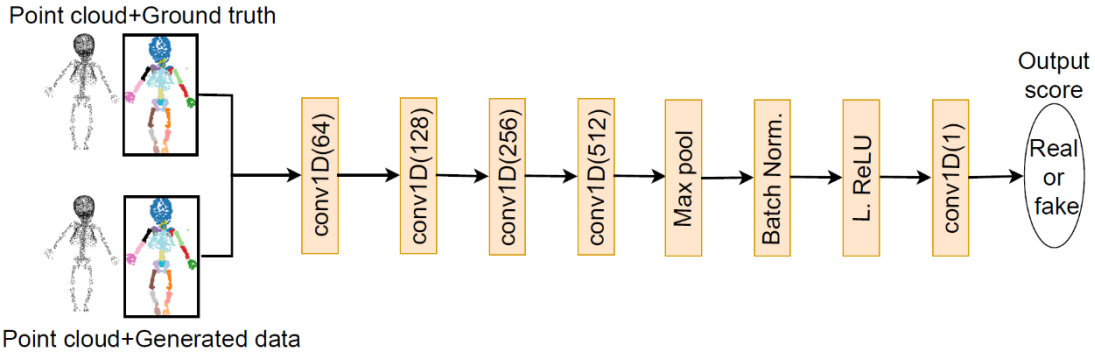


Figure 29. The structure of the discriminator model using convolutional neural networks in the present study.

Regarding the model hyperparameter, the Adam optimizer was used in both discriminator and generator with a beta of 0.5. The learning rate of the generator network begins at 0.001, then it is divided by 10 after every 50 iterations. In the discriminator, the learning rate was fixed at 0.001 for all loops. Figure 30 shows the change of learning rate according to steps from 1 to 200. We trained the GAN with different batch sizes of 1, 2, 8, 16, 32 and 64.

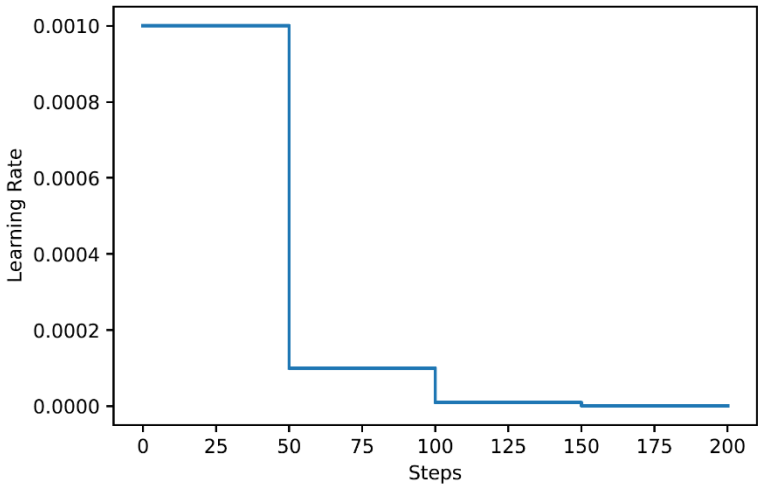


Figure 30. The learning rate schedule generated from our hyperparameter tuning process.

3.2.3 Pointwise convolutional neural network (Pointwise CNN)

Hua et al. [142] presented a convolutional neural network for semantic segmentation with 3D point clouds. The foundation of the method was a novel convolution operator known as pointwise convolution. This operation enabled the learning of features at individual points within a point cloud resulting in elegantly straightforward manner and fully convolutional networks tailored for the segmentation task. The input point clouds are supplied into 4 convolutional layers with 9 output channels and the ReLU activation function for each one. Then, all outputs are concatenated to feed into convolutional layers of 40 and *m* features to segment *m* parts. The activation function of the last layer is softmax. We trained the pointwise CNN with a learning rate fixed at 0.001 for all epochs. The used optimizer is Adam and batch size is 128. The pointwise CNN was convergent after 1000 epochs. Figure 31 illustrates the model of pointwise CNN for the part segmentation of newborn skeleton from 3D point clouds.

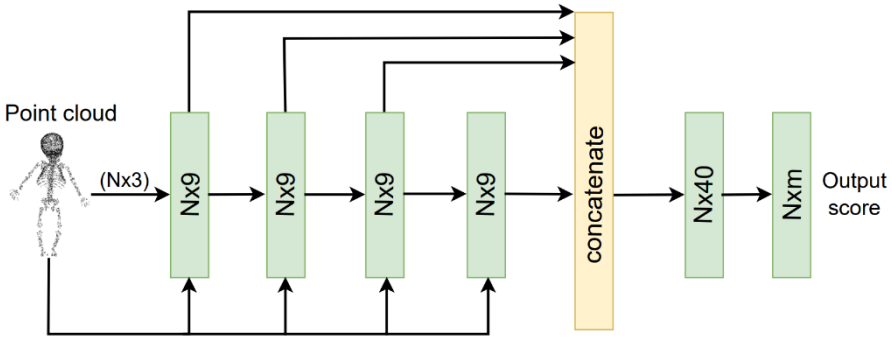


Figure 31. The pointwise convolutional neural network structure used in the present study for the part segmentation of newborn skeleton from 3D point clouds.

3.2.4 Performance metrics

We used different conventional metrics to evaluate the efficacy and performance of the part segmentation of the newborn skeletons. Several factors are involved in selecting an acceptable assessment metric, according to how the program operates [144]. This section contains some typical performance measures that we used for the present study.

3.2.4.1 Intersection over union (IoU)

The intersection over union metric is defined as the ratio of the overlapped area between the predicted segmentation and the ground truth segmentation to the union area between the predicted segmentation and the ground truth segmentation. It can be mathematically expressed using the following equation:

$$IoU = \frac{G \cap S}{G \cup S} \tag{22}$$

where S is the segmentation output and G is the gold reference data. This metric can be defined by using the Boolean notation as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{23}$$

where TP is the true positive, FP is the false positive and FN is false negative.

3.2.4.2 Dice similarity coefficient (DSC)

The Dice coefficient analyzes a spatial overlap between the segmentation outcome and the ground truth. Note that a greater value indicates a better segmentation performance. A DSC score varies from 0 to 1, with 0 implying that there is no spatial overlap between two binary segmentation sets and 1 representing the total overlap. The DSC formula is mathematically defined as follows:

$$Dice = \frac{2|G \cap S|}{|G| + |S|} \tag{24}$$

When apply to Boolean data, the equation becomes as follows:

$$Dice = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{25}$$

where TP is the true positive, FP is the false positive and FN is false negative.

3.2.4.3 Accuracy score

The accuracy score is considered as a rand index to quantify the proportion of accurate predictions, which encompasses both correctly predicted positives and negatives divided by the total performed predictions. The mathematical equation of this metric is expressed as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

where TP is the true positive, FP is the false positive and FN is false negative.

3.2.4.4 Precision and recall metrics

The precision metric is defined as the ratio of true positive predictions to the total number of predicted positives. In mathematical terms, the precision metric is calculated using the following formula:

$$Precision = \frac{TP}{TP + FP} \tag{27}$$

where TP is the true positive and FP is the false positive.

The recall metric is a measure used for assessing the performance of a machine learning model, especially in binary classification issues. It is also related to the sensitivity or true positive rate. The recall metric is given as the ratio of true positive predictions to the total number of actual positives (both true positives and false negatives) as follows:

$$Recall = \frac{TP}{TP + FN} \tag{28}$$

where TP is the true positive and FN is false negative.

It is worth noting that the precision and recall metrics are often considered together and there is a trade-off between them. The precision metric highlights the accuracy of positive predictions, whereas the recall metric concentrates on the capacity to capture all relevant instances of a class. Combining these two metrics allows us to comprehend the advantages and disadvantages of the proposed model.

3.3 Computational results

3.3.1 GAN-based segmentation outcomes

The best GAN model was with a batch size of 1 looped 150 steps that returned a mean IoU of 93.68% ± 7.37%, a mean Dice of 96.56% ± 4.41% and a mean accuracy score of 96.72% ± 3.56% for predicted segmentation on the validation set. The second good one was the GAN model with a batch size of 2 obtained a mean IoU from 78.97% to 89.09% for 50 to 200 iterations, respectively. Training the GAN model with a batch size of 32 returned the lowest results with a mean IoU of 76.09% after 50 steps and just reached to a mean IoU of 84.46% after 200 loops. All average performance metrics of predicted results on the validation set by our GAN models trained with different batch sizes and iterations are depicted in Table 1. In the case of batch size of 8, the accurate prediction on the validation set slightly increased from 79.96% to 85.35% for the mean IoU after 200 iterations while the Dice score increased from 88.47% to 91.66% and the accuracy changed from 88.56% to 92.02%. The GAN model with the batch size of 64 obtained the predicted result not significantly different after 50 iterations and 200 ones with 85.37% and 85.40% for the mean IoU, respectively. The GAN model using the batch size of 16 acquired the highest mean IoU of 85.29% at the iteration of 150 for the prediction on the validation set. In training a GAN model, setting a larger batch size could potentially degrade the performance as it may lead to an imbalance effect during the initial training. The discriminator might be given numerous samples that lead to dominating the generator and thereby adversely impacting the training process and overfitting. In this study, using a batch size of 1 showed the best outcome. Figure 32 illustrated the mean IoU evolution of predicted results on the validation set using our GAN model with different batches over different steps. Note that the training time for each step was about 1 minute on Google Colab GPU.

With the best GAN algorithm, we obtained the best predicted newborn skeleton part segmentation with 99.63%, 99.81% and 99.98% for the means of IoU, Dice and accuracy score, respectively. Meanwhile, the worst case returned the mean IoU of 65.81%, mean Dice of 79.37% and mean accuracy score of 80.03%. Figure 33 and Figure 34 illustrated the best and worst case of bony part segmentation from 3D point clouds using our GAN model.

The segmented results on the training set, testing set and validation set using GAN with a batch size of 1 were described in Table 2. The prediction on the training dataset returned a very high mean IoU of 99.22 ± 0.87%, a mean accuracy score of 99.64 ± 0.24%, and the same Dice, precision and recall metrics of 99.61 ± 0.45 %. On the testing set, the segmented output was 92.52 ± 8.13 % for the mean IoU, 96.03 ± 4.96 % for the mean accuracy and 95.45 ± 6.78% for the mean Dice, precision and recall metrics. Note that the high score and similarity of precision and recall metrics point out that the model is very accurate and not missing many positive instances. Moreover, they share similar values with the Dice coefficient, which confirms how precise and robust our used model is. Note that the Dice coefficient and IoU metric have specific characteristic of penalizing false positives, which are especially crucial in datasets with substantial class imbalances.

Table 1. The means (in %) of the intersection over union (IoU), Dice similarity coefficient (Dice), accuracy score (Acc) of predicted segmentations on the validation set using our GAN model with different batches according to different iterations of 50, 100, 150 and 200.

Iteration	50			100			150			200		
Batch	IoU	Dice	Acc	IoU	Dice	Acc	IoU	Dice	Acc	IoU	Dice	Acc
1	93.60	96.54	96.70	93.67	96.56	96.72	**93.68**	**96.56**	**96.72**	93.66	96.55	96.71
2	78.97	87.96	88.01	88.46	93.66	93.98	88.73	93.81	94.14	89.09	94.02	94.31
8	79.96	88.47	88.56	83.69	90.66	91.03	84.08	90.89	91.58	85.35	91.66	92.02
16	84.67	91.36	91.51	85.06	91.58	91.97	85.29	91.75	92.15	84.42	91.22	91.53
32	76.09	86.10	86.37	82.56	90.10	90.63	80.48	88.88	89.13	84.46	91.22	91.53
64	85.37	91.98	92.36	85.18	91.88	92.26	83.86	91.11	91.32	85.40	91.98	92.36

Table 2. Comparison of mean ± standard deviation (in %) of the IoU, Dice, accuracy, precision and recall metrics of the prediction segmentation on the training set, test set and validation set using pointwise CNN and GAN models.

		IoU	Dice	Accuracy	Precision	Recall
Pointwise CNN	Training	75.50 ± 0.75	86.04 ± 0.49	88.04 ± 0.82	86.04 ± 0.49	86.04 ± 0.49
	Test	71.12 ± 5.69	82.26 ± 3.86	83.65 ± 4.01	82.26 ± 3.86	82.26 ± 3.86
	Validation	72.30 ± 5.10	83.82 ± 3.44	84.81 ± 3.25	83.82 ± 3.44	83.82 ± 3.44
GAN (batch 1)	Training	99.22 ± 0.87	99.61 ± 0.45	99.64 ± 0.24	99.61 ± 0.45	99.61 ± 0.45
	Test	92.52 ± 8.13	95.45 ± 6.78	96.03 ± 4.96	95.45 ± 6.78	95.45 ± 6.78
	Validation	93.68 ± 7.37	96.56 ± 4.41	96.72 ± 3.56	96.56 ± 4.41	96.56 ± 4.41

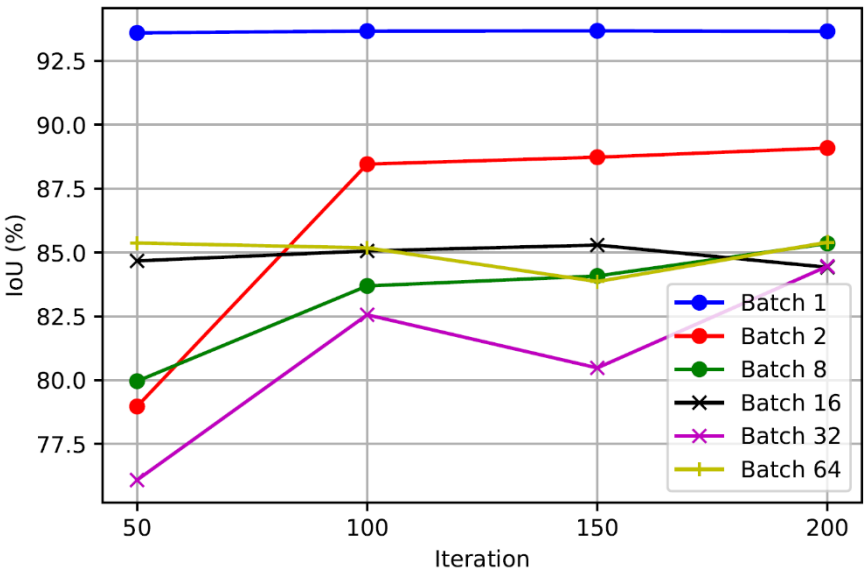


Figure 32. The evolution of the obtained intersection over union (IoU) metrics on different batches and over different iterations using our proposed GAN model.

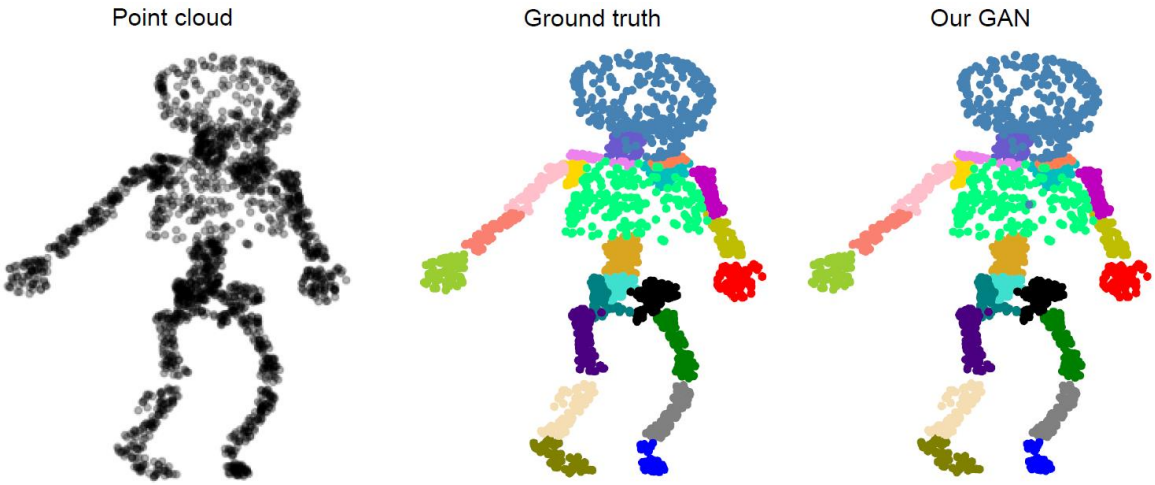


Figure 33. The best predicted skeleton part segmentation by our GAN method with a mean IoU of 99.63%, a mean Dice of 99.81% and a mean accuracy score of 99.98%.

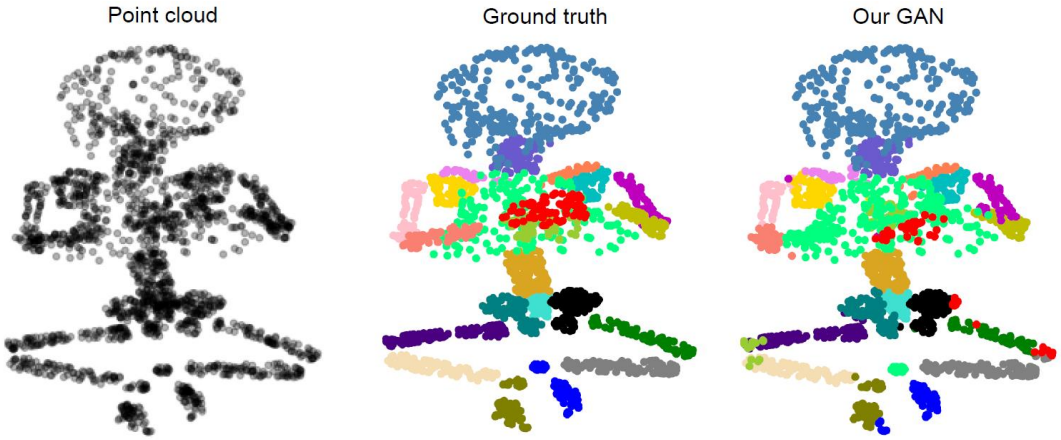


Figure 34. The worst segmentation case by our GAN with a mean IoU of 65.81%, a mean Dice of 79.37% and a mean accuracy score of 80.03%.

3.3.2 Pointwise CNN segmentation outcomes

For Pointwise CNN method, we trained the pointwise CNN model with the batch size of 128, learning rate of 0.001 and Adam optimizer. The number of epochs was set up as 1000. It took very quick to train on Google Colab GPU, just around 1 second for one epoch. The predicted segmentation on the training set returned a mean IoU of 75.50% ± 0.75%, a mean Dice of 86.04% ± 0.49%, a mean accuracy of 88.04% ± 0.82%, same mean precision and recall of 86.04% ± 0.49%. On the testing set, the model predicted bony segmentation with 71.12% ± 5.69% for the mean IoU, 83.65% ± 4.01% for the mean accuracy and 82.26% ± 3.86% for the same mean Dice, precision and recall metrics. The segmented results on the validation set obtained a mean IoU of 72.30% ± 5.1%, a mean Dice of 83.82% ± 3.44% and a mean accuracy score of 84.81% ± 3.25%. The best case resulted 86.31% for the mean IoU, 88.75% for the mean Dice and a mean accuracy score of 89.37%. The worst prediction showed a mean IoU of 45.10%, a mean Dice of 62.15% and an accuracy score of 62.88%. The visualization of the best segmentation case and the worst one is shown in Figure 35 and Figure 36, respectively.

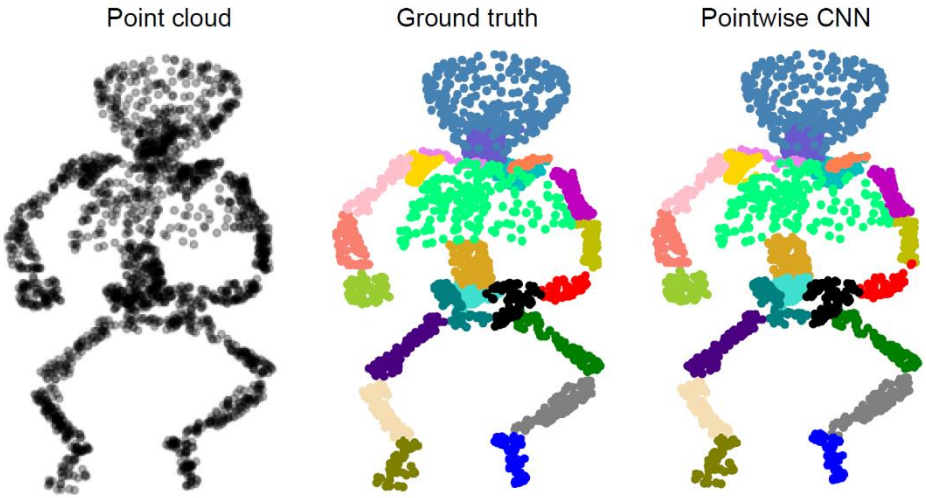


Figure 35. By using the pointwise CNN method, the best result was obtained with a mean IoU of 86.31%, a mean Dice of 88.75% and a mean accuracy score of 89.37%.

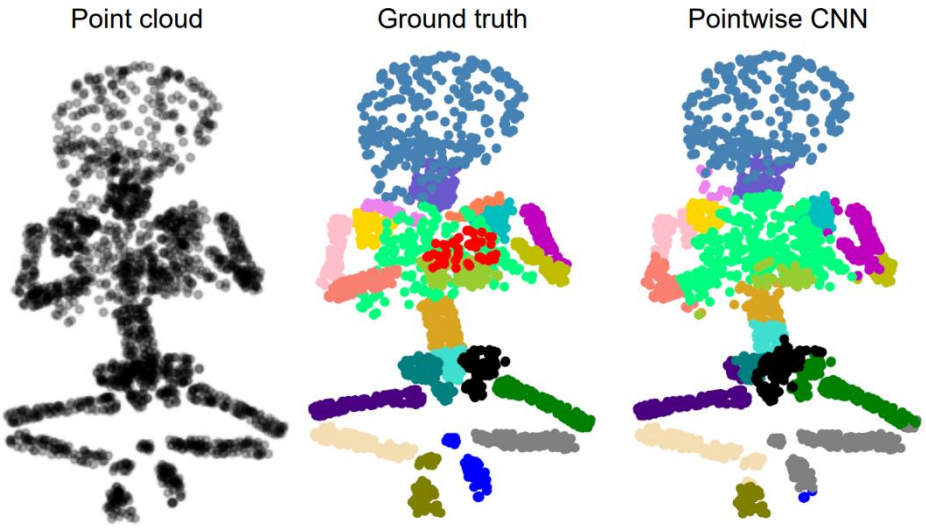


Figure 36. The worst segmentation case from the used pointwise CNN method with a mean IoU of 45.10%, a mean Dice of 62.15% and an accuracy score of 62.88%.

We summarized the performance of our used GAN and pointwise CNN methods in segmenting 23 different body parts of the newborn skeleton in Figure 37. The GAN method achieved a mean IoU of around 85% for the R-scapula segment and around 99% for skull segment while the pointwise CNN returned the worst segmentation case with a mean IoU of 42% for the R-clavicula segment and also obtained the highest mean IoU at skull segment with a mean IoU of 95%.

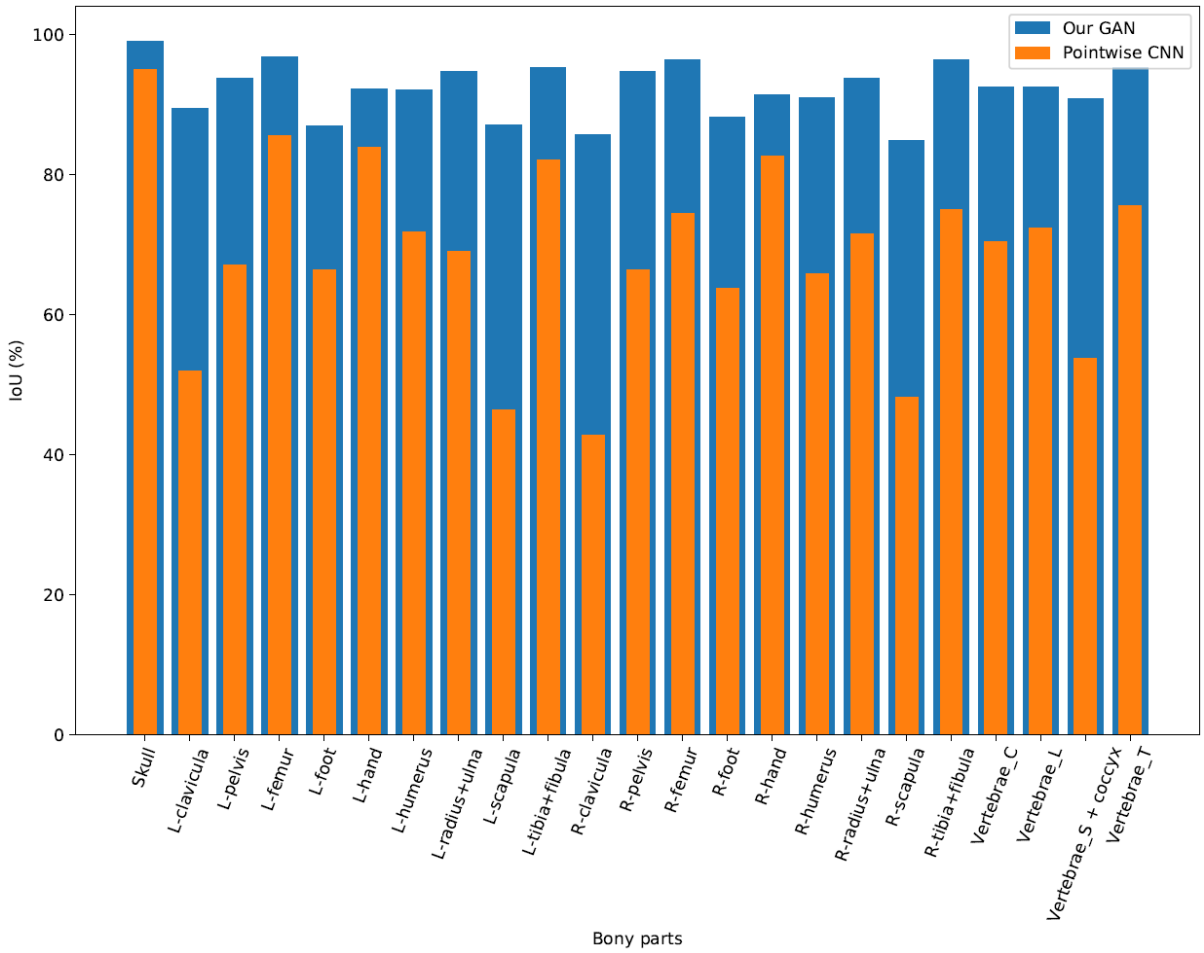


Figure 37. The performance comparison using the IoU metric between the GAN and pointwise CNN methods for segmenting each bony part from the full skeleton.

3.4 Discussion

The foetus descent through the birth canal during the childbirth process has a complex pattern. The successful vaginal delivery depends strongly on this kinematic process. Numerical modeling has been commonly used to reproduce and describe this process for a better understanding due to ethical issues of the experimental studies [23], [126], [145]. However, the foetus model is still simplified, especially on the description of articulated joint mechanics. To develop a more detailed articulated skeleton model, the segmentation and reconstruction of separate 3D segments of the fetal skeleton are still challenging and time-consuming. In particular, cleaning and isolating a large number of segments from the whole 3D skeleton demands a substantial work. The task of labelling 23 segments in this study, for example, consumes about 30 to 50 minutes by a skilled operator using 3D Slicer software. In this work, we proposed a generative adversarial network to accurately and quickly segment bone

components, just in 5 seconds. Our GAN model showed also that the segmentation performance is higher than that of the pointwise convolutional neural network. However, the pointwise CNN model took cheaper training time than the GAN model.

Within the context of 3D point cloud processing, Achlioptas et al. [146] first presented GAN models to reconstruct and interpolate objects and human motions from 3D point clouds. The model was coupled with the auto-encoder network, which learns to reproduce point cloud inputs, including an encoder net and a decoder net. The encoder used 5 layers of 1D-convolutional and a batch normalization layer. The decoder used 3 fully connected layers to produce outputs. By training the auto-encoder first, then GAN models were trained in the latent space of the auto-encoder. The generator model included 2 fully connected layers and a ReLu layer. The discriminator also consisted of 2 fully connected layers but with a sigmoid one. The limitations of the approach were that the auto-encoder tended to overlook highly frequent features or fail to reconstruct rare geometries. Moreover, the GAN had difficulty in creating realistic-looking forms and the synthesis of unrealistic results. Yu et al. [147] presented a network, which was relied on PointNet and GAN, for 3D point cloud inpainting. The model produces the missing portion straight from the defective point cloud. Both the generator and the discriminator in this method were based on the PointNet model. Whereas, we built our discriminator using convolutional neural networks with fewer layers (i.e. 5 1D-convolutional layers), a max-pooling and a batch normalization layer. The discriminator does not need to distinguish on high-resolution images because here is a study of a binary segmentation problem. Thus, we execute this discriminator convolutionally over the inputs, aggregate the responses and average them to generate the final outputs. This approach can reduce the system memory and computational cost compared PointNet scheme. We harnessed the benefits of the T-net structure, which enhanced the feature extraction from the training data and facilitated the adjustment of rotation angles, in the development of the generator. We found out that the GAN model with the batch size of 1 give the best segmentation outcomes and cheaper memory space.

To extract the human skeleton from 3D point clouds, Qin et al. [148] presented a convolutional neural network-based (CNN) method. The framework comprised two phases. First, ambiguous points were eliminated from the raw dataset. Second, the interest point clouds were lowered toward the matching joint points. The first network was trained on 6000 epochs with batch size of 16, the second one was trained on 4000 epochs with batch size of 8. The initial learning rate was 0.001 and the Adam optimizer was used. On the raw point clouds, the results returned a mean IoU range of 76% to 81%, while interest point clouds outputted a mean IoU range from 92.6% to 98.7%. Takmaz et al. [149] created a synthetic training dataset for virtual people in accurate 3D interior scenes and proposed a transformer-based model, named Human3D, which unifies the performance of 3D body part segmentation in point clouds. The model included a sparse convolutional filter backbone coupled with a transformer decoder. Within the anticipated human instance mask, the body-part masks, based on two-level searches, are repeatedly improved using multi-scale point features. The model was trained on 36 epochs. The Adam optimizer was used with a learning rate schedule of maximal 0.0001 and a batch size of 4. The segmentations of 15 body parts obtained the mean IoU of 69.9%. Zang and Wang [138] employed PointNet++ [150] to segment cervical spine part from bone tissue structures, which were segmented from CT images. Then, applying a combination of edge and concave points to delineate the boundary, which overlaps between segmented vertebrae. The segmentation outcomes achieved an accuracy score of 96.15%. In our study, the best GAN model with batch size of 1 and 150 epochs produced an IoU of 93.68% ± 7.37%, a Dice of 96.56% ± 4.41% and

an accuracy score of 96.72% ± 3.56% for segmenting 23 bony segments of the newborn 3D skeleton. In fact, the use of modern deep learning approaches (e.g. CNN, transformer-decoder model, GAN) showed great potential applications for 3D human skeleton segmentation with a high accuracy level. However, the model interpretability remains a challenging issue when applying such approaches in the medical context to help the medical doctors and professionals in their decision-making process. The present study limited the use of the GAN method on the reconstruction of the detailed articulated skeleton for childbirth simulation purpose.

Since point clouds are chaotic and unstructured, conventional convolutional neural networks (CNNs) cannot be implemented directly into them. CNNs demand that data possess a specific orientation, meaning that CNNs do not inherently exhibit invariance to changes in picture orientation. PointNet, considered as a groundbreaking method, processes point clouds by acquiring spatial representations for individual points, consolidating these learned representations into features, then supplying them to tasks such as classification and segmentation. PointNet employs shared MLP layers and symmetric pooling functions to generate global features. Numerous networks have been created by using the PointNet framework due to its straightforwardness and powerful capacity to represent data. Wang and Lu [151] proposed the VoxSegNet model for the part segmentation of 3D objects from voxelized data. The method performed well for part segmentation of many objects such as airplanes with an average IoU of 86.2%, and 92% for chairs. Yu et al. [152] developed the first model for hierarchical segmentation of 3D objects, called PartNet. The segmented results by PartNet showed an average IoU of 87.8% for part segmentation of airplanes, and 91.9% for chairs.

However, the limitation of the PointNet deals with the failure to capture the local context information between points because features have been learned individually for each point. In general, point-based methods cannot represent explicit neighboring information. Thus, many methods need to rely on costly neighbor searching algorithms. This restricts the effectiveness of existing approaches. Nonetheless, the recently suggested point-voxel joint representation can be promising in the future study. Although GAN's recent advancements have brought fresh concepts for 3D point cloud segmentation, it still has certain practical limitations. We assessed the ability to implement the GAN-based model rather than its efficacy in clinical applications. There are still a few challenges, including the training set up complexity and cost as well as non-convergence issues.

While typical CNNs segment data on a per-pixel or per-voxel basis by optimizing segmentation metrics such as dice coefficient and intersection over union, GANs additionally penalize segmentation outcomes lacking visual realism. The adversarial loss, which the discriminator introduces, offers a creative technique to include unlabeled samples in the training process and enforce higher-level consistency. The discriminator guides the outcomes of the generator toward the actual distribution thereby improving the generator's work by refining what it generates. In the present study, we build the discriminator using CNN architecture that accepts the combined setup of multiple input variables, thus higher order possibilities are integrated into the network. This allows representing the geometric disparity between the prediction and real through the trainable model, rather than relying on heuristic clues.

From the segmentation performance and visualization aspect, we found out that the accuracy of the outcomes depends on the fetal postures. When the subjects were in physiological position, a higher accuracy level was achieved. In the bad segmentation scenario, the arms of babies

rested above their chests. This posture made the model harder to distinguish the hands from one another and the forearms and hands from the coasts. This finding is interesting given the goal of our segmentation method to precisely segment the various bony segments from 3D point clouds collected via MRI or 3D ultrasound scan of developing fetuses. This strategy could be used for creating personalized childbirth simulations models. It is worth noting that the data collection involves unborn fetuses and these subjects are typically in a fetal position within the maternal uterus. Moreover, the difficulty of distinguishing between fused organs such as vertebrae is also a limitation of the present study. Because of the significance of the cardinal motions during natural childbirth, creating an articulated musculoskeletal model of the fetus for delivery simulations necessitates the precise spine articulation, particularly for the cervical vertebrae. Thus, future study should be investigated to solve this issue. Finally, the small sample size is also a limitation of the present study due to the challenging acquisition of fetal data. Thus, more data will be added when available to enhance the findings. The use of transfer learning strategy [129] from the adult data, which is largely available, to the fetal data could be also a potential solution. However, the bone shape discrepancies between adults and fetuses should be elucidated before applying this strategy.

Chapter 4

A Novel Deep Learning-Driven Approach for Predicting the Pelvis Soft-Tissue Deformations toward a Real-Time Interactive Childbirth Simulation

The dynamics of soft tissue are pivotal for understanding the mechanical behavior of the human body. Presently, numerical methods employing finite element modeling and mass-spring frameworks are employed to approximate the dynamic properties of biological soft tissues. Despite their utility, these approaches are encumbered by significant computational expenses, largely stemming from the necessity of mesh configurations in formulating dynamic equilibrium equations and encountering challenges related to unstable convergence. Addressing these issues is crucial for advancing our comprehension of soft tissue dynamics and enhancing the efficiency of computational simulations in biomechanical studies.

This chapter introduces a methodology rooted in the deep learning paradigm to forecast the deformation of uterine soft tissues. Specifically, the utilization of Long Short-term Memory (LSTM) neural networks and deep neural networks (DNN) is proposed to handle the intricate high-frequency oscillation signals inherent in such biological systems. Various learning strategies, encompassing approaches both with and without data dimension reduction, will be explored to optimize performance. To facilitate model development and evaluation, a simulation-based database is constructed utilizing the HyperMSM model, serving as a comprehensive resource for training and testing purposes. This innovative approach holds promise for enhancing our understanding of uterine dynamics and advancing predictive capabilities in childbirth simulation.

4.1 Introduction

Biological soft-tissue dynamics plays an essential role in the mechanical functions of the human body [153], [154]. Biological soft tissues commonly exhibit a nonlinear behavior when interacting with other tissues and external devices. For example, skeletal muscle excitation and contraction contribute into the dynamic movements such as walking or running. Another example relates to the deformation of the uterus organ to deliver the fetal body during childbirth [155]. In particular, when developing the interactive childbirth training tool with man-machine interaction, the prediction of the soft tissue deformation is of great clinical relevance to optimize the training gestures and design novel delivery devices (e.g. forceps with adaptive control using soft tissue feedback). At the moment, there is no reliable experimental protocol to measure the soft tissue dynamics in non-invasive and *in vivo* conditions. Medical imaging techniques such as dynamic magnetic resonance imaging (dMRI) [156] or magnetic resonance elastography (MRE) [157] could be used. However, only limited information (e.g. 2D mobilities or local material properties) could be extracted. Numerical modeling becomes an alternative solution to estimate the biological soft tissue dynamics. The finite element method has been commonly used to estimate the soft tissue deformation and stress under internal and external mechanical

loadings [158]. Besides this, mass-spring model (MSM) formulation has been also applied to achieve a fast estimation of the soft tissue dynamics [159]. However, these numerical approaches use a mesh configuration in the formulation of the dynamic equilibrium equation to solve the problem. This still leads to important computational cost, especially for solving the principle of virtual displacement equation in the framework of the finite element theory. Note that some studies proposed potential improvements of finite element methods to speed-up the computational time. For example, the Kalman filter method, which was introduced by Song et al. [160], reduces nonlinear order. Xie et al. [161] proposed a new constrained finite element method that approximates undetermined applied forces to compute the deformation of soft tissue. Furthermore, the convergence stability is also a challenging issue for these approaches to obtain a reliable estimation outcome [153], [154]. In fact, the integration of developed models into an interactive tool remains a scientific and technological challenge.

In addition, machine learning has been the state-of-the-art approach applied to numerous domains such as natural language processing, computer vision and medical diagnosis [162], [163], [164]. The ever-increasing volume of information is now accessible and the development of innovative algorithms and high-performance computing systems are the reasons why machine learning has been widely employed recently. A well-known type of machine learning approach is deep learning, which relies on a deep neural network for learning data representations rather than specified task algorithms. This approach has been found to be very effective in extracting detailed representations of complicated processes. The deep learning architecture commonly includes a feedforward neural network with weighting features and nonlinear transform elements, so-called units. These units are set up into layers, with the input data passing through them multiple times before it becomes output [165], [166]. The applications of deep learning to predict physical quantities in computational mechanics have been significantly increasing in recent years. Nguyen et al. [167] proposed the long short-term memory (LSTM) model and multilayer neural networks to forecast the crack propagation of concrete beams and in hydraulic fracturing. Combining LSTM with a hidden Markov model to predict the growth of fracture was introduced by Nguyen-Le et al. [168]. Kollmann et al. [169] presented a convolutional neural network model for topology optimization of metamaterials, which used the energy homogenization method and periodic boundary conditions to create data. Haghighat et al. [170] studied the application of physics-informed neural networks to linear elasticity problems, then performed on nonlinear elastoplasticity with data generated from finite element methods and isogeometric analysis. Moreover, the uses of deep learning in fluid mechanics can be found in [171], [172] and for material science presented in the literature [173], [174]. In the discipline of biomechanics, the models based on deep learning approach have been widely utilized for segmentation works [175], [176], [177]. To classify magnetic resonance spectroscopy signals in order to detect brain tumors, a model based on LSTM was also studied [178]. Regarding the estimation of mechanical quantities (e.g. displacement, strain and stress fields), Zaroug et al. [179], [180] used LSTM to extrapolate the kinematics of lower extremity motions during walking. Khadem and Rey [180] combined the direct numerical simulation and LSTM to analyze the nucleation and growth of collagen-based biomaterials. Dao [129] employed an LSTM network to predict the skeletal muscle forces from joint kinematics data during a walking period. Recently, this approach has been also applied to predict skeletal muscle stress during a dynamic contraction behavior [130]. In fact, the deep learning approach and particularly the LSTM neural network with gate structure ensuring the efficient dependence learning between input data set has shown its great capacity to deal with time series data with nonlinear and high-frequency oscillations.

Innovative data-driven computing approaches have introduced a novel way of advancing computational mechanics. The rapid utilization of data-driven computing has been focused on

resolving challenges related to computational material science, for example, the use of experimental results to substitute constitutive equations by Kirchdoerfer and Ortiz [181]. The development of data-driven methodologies for multiscale modeling has been shaped by the works of Mou et al. [182] that presented a data-driven framework for creating a reduced order model, which is built upon the hierarchical arrangement of the variational multiscale approach. This framework leverages data to enhance the accuracy of the reduced order model while maintaining a reasonable level of computational resources. In the framework of dynamic simulations, Kirchdoerfer and Ortiz [183] developed data-driven computing methods that encompass the inclusion of time integration within both distance-reduction and entropy-enhancement approaches. Recently, data-driven methods have been employed to soft tissue mechanics. Santesteban et al. [94] proposed a data-driven model that acquired input of shapes of human body and motion using recurrent regression to predict deformations of body surface soft tissues. A data-driven approach called the local convexity data-driven model was introduced by He et al. [184] to simulate the mechanical behavior of the posterior leaflet of a porcine heart. The efficiency of the framework was examined through the utilization of different combinations of biaxial and pure shear training protocols, demonstrating its performance. Tac et al. [185] provided data-driven modeling using deep neural networks to predict strain energy and its derivative functions while capturing the anisotropic mechanical characteristics of porcine and murine skin. Data-driven modeling can be an alternate approach for simulating complicated biological tissues when adequate data is accessible.

In summary, deep learning has metamorphosed different engineering fields with more accurate prediction and faster computation capacities. However, this approach is still not investigated for the pelvis soft tissue deformation prediction within an interactive childbirth simulator. In this present work, we develop and train different deep learning architectures (classical deep neural network and LSTM) and different learning strategies (without and with data dimension reduction) to predict the deformation of soft tissues for childbirth simulation. The proposed models analyze the relationship between dynamic displacement with node position, velocity and external force. The objective is to provide a real-time estimation with stable prediction outcome.

4.2 Materials and Methods

4.2.1 Novel workflow and mathematical background of the soft tissue deformation prediction

To estimate the soft tissue deformation in real-time and stable condition, a novel workflow was developed and shown in Figure 38. First, a simulation-based database was generated from a childbirth simulation performed using our recently developed HyperMSM model [159]. Next, different deep learning architectures and learning strategies were designed, implemented and evaluated. Finally, the best trained machine-learning model will be integrated into a runtime interactive simulation of the physiological childbirth procedure.

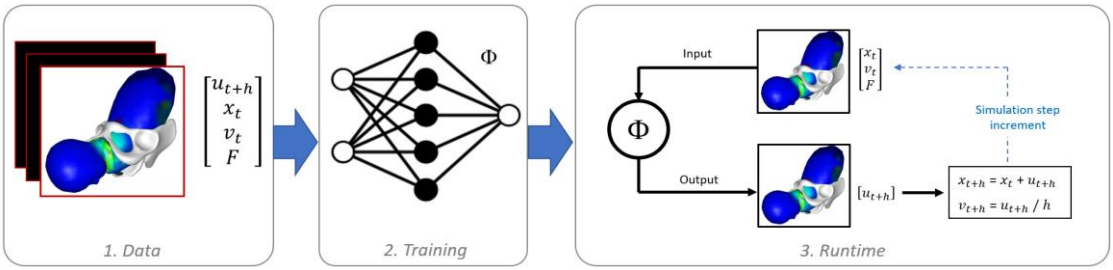


Figure 38. The proposed data-driven workflow coupling the database generation, deep learning network implementation and deployment.

A 3D-model usually consists of a closed 3D-surface made of nodes and facets. A deformable 3D-model is such a model that exhibits deformations as a response to internal and external mechanical loadings. A dynamic deformation simulation is time-dependent and involves a discretization of the time domain and consists of finding the new nodal positions at each time-step. The prediction problem can be thus rewritten as follows:

$$x_{t+h} = f(x_t, v_t, F, h, material\ properties) \tag{29}$$

where the current nodal position vector at each simulation instant is noted x_t, whereas the new one to be calculated is noted x_{t+h}. Similarly, we note v_t the current nodal velocity vector, and v_{t+h} the updated one. F is the external foces vector and h is the simulation time-step. It is also possible to work with the nodal instant displacement vector u_{t+h} instead of x_{t+h} knowing that $x_{t+h} = x_t + u_{t+h}$. Hence, we get the following reformulation of the problem as follows:

$$u_{t+h} = \varphi(x_t, v_t, F, h, material\ properties) \tag{30}$$

Equation 30 can be interpreted as follows: to predict the nodal displacement of a deformable model within a time-step, we need to know the time-step value (h), the material properties of the model, the external forces (F) applied to the nodes, the current nodal (x_t) position and the current nodal velocity (v_t).

4.2.2 Childbirth model and learning database generation

The learning database was generated through childbirth process simulations. Our training process takes raw time-series data consisting of vertex positions, velocities, and interactive forces applied to the nodes, and produces predicted nodal displacements. This makes a wide range of physics-based simulation methods applicable for gathering the necessary data for our approach. In this study, we opted to utilize our in-house HyperMSM simulator to conduct simulations and compile the database, since HyperMSM has demonstrated its ability to rapidly simulate the biomechanical characteristics of soft tissues with a satisfactory level of physical accuracy [23], [159]. As for the childbirth model used in these simulations, we employed a previously developed generic model [23]. The model consists of a fetal body, pelvic soft tissues combining the floor muscles, vagina and uterus, and the bony structure represented by the pelvis (Figure 39). In reality, this constitutes a simplified model that does not mirror the anatomical complexities of the female pelvis system; nonetheless, it is deemed sufficient for the immediate objective of this study.

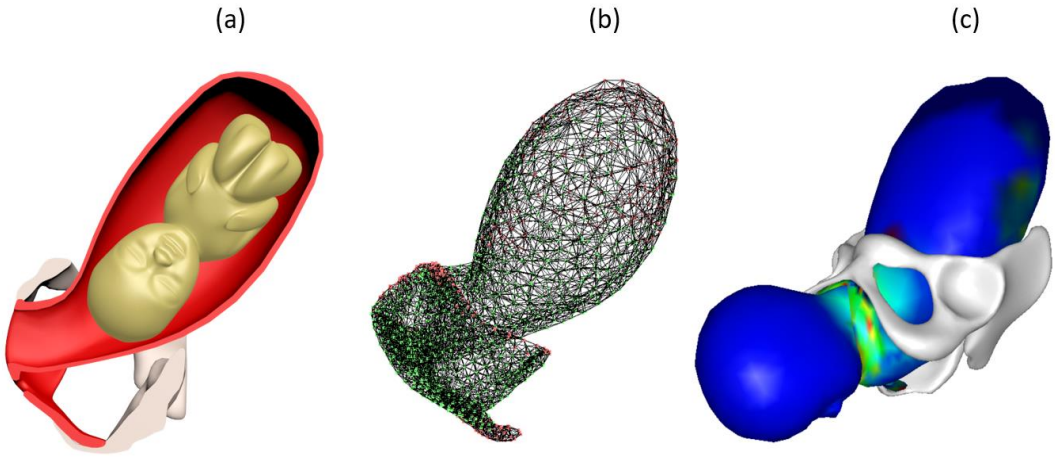


Figure 39. Our childbirth model. (a) shaded 3D model sliced for visualization. (b) the HyperMSM model of the pelvic soft tissues, the red nodes are the bodes fixed in the space. (c) One frame of the HyperMSM simulation of the physiological chilbirth scenario.

Material properties utilized in our models are drawn from existing literature. Lepage et al. [186] provided estimations of mechanical attributes concerning pelvic organs in both pregnant and non-pregnant women, focusing on Young's Modulus—a parameter relevant to linear elasticity. Given the design of HyperMSM to portray the Neo Hookean model, renowned for its simplification of soft tissue mechanics [23], the computation of Neo-Hookean parameters from linear elasticity parameters is feasible, using the well-known equivalence relationship between these two models for small deformations. Consequently, for the pelvic soft tissues model, we assigned C_{10} = 10.34 kPa and D^{-1} = 100 kPa, values derived from E=60 kPa, as stated by Lepage et al. [186] for the pelvic floor muscle of a pregnant female, accompanied by a Poisson ratio of 0.45 [187], [188]. Regarding the fetal model, we adopted C_{10} = 70 kPa [189] and D^{-1} = 66.66 kPa. A density of 1g/cm³ was uniformly applied to all of our models, as it aligns with the established soft tissue density within this range [190].

The HyperMSM models of the deformable objects (pelvic soft tissues and baby) were created through volumetric tetrahedral meshing, achieved through *Abaqus CAE*. The final mesh of the pelvic soft tissues comprises 6874 tetrahedral elements, thus yielding a HyperMSM model equipped with 10,790 springs and 2103 nodes. Notably, 450 nodes were subject to a boundary condition, constraining their displacement (Figure 39b). This encompasses nodes located within the interconnected region of the soft tissues and the pelvis bone, as well as nodes corresponding to ligament insertion sites. Additionally, the terminus of the uterus was restrained to avert undesirable uterine displacements or collapse during the simulation. As for the fetal model, a mesh resolution involving 6274 tetrahedral elements was implemented, resulting in a fetal HyperMSM model comprising 8269 springs and 1382 nodes. The fetal nodes, unlike their pelvic counterparts, remain unconstrained, as the fetus is free in the space. Notably, the geometrical representation of the pelvic bony structure takes the form of a rigid entity, modeled as a three-dimensional triangular surface that remains entirely immobilized in space.

In reality, the fetal movements during the labor are induced by the uterine contractions (causing the intra-uterine contraction forces) and the interactions of the fetal head with the pelvic anatomy which causes what is known as "fetal cardinal movements". To simplify this, we simply applied to body force to the fetal model whose direction changes according to a predefined trajectory that mimics the curve of Carus [191], along with applied torques at each stage to simulate the rotations of the fetal head at the different stages of the cardinal movements

[192]. This *force-based trajectory* was employed instead of an imposed displacement along a trajectory, to preserve the significant role played by the resistance exerted by the pelvic soft tissues on the fetus, and also to enable interactions between the fetus and external interactive forces. The use of an imposed displacement would negate these crucial dynamics. Contact between the baby and the soft tissues was modeled using a node-to-surface discrete collision algorithm coupled with elastic penalty forces. Further elaboration on the executed dynamics of our model can be found in [23].

To generate the database, we conducted several HyperMSM simulations of the childbirth process, maintaining constant material properties across all simulations. The simulation time-step was also constant and set to 2 ms for all cases. In fact, HyperMSM uses the Projective Dynamic (PD) algorithm to solve the time integration. While many algorithms can support variable simulation time-step, PD necessities a constant one all over the simulation [193]. The selection of the time-step value plays a critical role in dynamic simulations: a smaller time-step ensures greater simulation stability, while a larger time-step speeds up the simulation. In our application, we opted for a 2ms time-step to strike a balance between speed and stability. Thus, by treating the material properties and time-step as constants in Eq. 30, a database consisting of a set of records formed of vectors x_t, v_t, F, and $u_{t+h}$was generated. The first three vectors will form the input set of the deep learning model, whereas u_{t+h} will be the output. We only consider in this stage the surface points that are not constrained in the 3D-space. The number of the surface nodes of our model is 1653, and therefore, the size of each of the vectors to be collected is 3x1653. Thus, multiple HyperMSM childbirth simulations were performed in which the data was captured at 100 frames per second. At each simulation, we applied random external forces to the surface of the pelvic floor muscles to consider the clinical intervention forces that may be interactively applied to this area. Thus, the model of the pelvic soft tissues is subject to these randomly generated external forces as well as the contact forces induced by the collision with the baby model. Eventually, we acquired a database of 15396 frames for training. The test database, similarly, generated 18633 frames with 6 simulations.

4.2.3 Deep learning architectures and learning strategies

4.2.3.1 Long Short-term Memory (LSTM)

The accurate prediction of complex time series data requires a specific neural network. One of the most well-known models used to solve this problem is the recurrent neural network (RNN). RNN was derived from feedforward neural networks, which only transfer information in one direction from the input layer to the output layer via the hidden layers. RNN features loops inside its network, which enable it to preserve information of previous inputs in a not prior fixed time, hence this network considers the weights of the current and the past inputs as well. An extension model of RNN is the LSTM network, which was proposed by Hochreiter and Schmidhuber in 1997 [35] to address the problem of vanishing gradients. A significant finding in the LSTM architecture was the integration of non-linear and data-dependent controls to an RNN that helps the loss function overcome the vanishing gradients problem. Assuming that the unfolded LSTM units are performed as in Figure 40 for k time steps. The LSTM unit at the step t uses the input x_t, the previous output h_{t-1}, and the cell state signal c_t to compute the output signal h_t at this present step. Figure 41 shows the architecture of a LSTM unit. A common LSTM unit contains three control gates that are forget, input and output. These gates decide which information is to store or delete from a cell state. The forget gate was attached to the LSTM structure by Gers et al. 1999 [194] to remove unimportant information in the cell, the input gate restricts the amount of new information entering the cell and the output gate is to control the information used for the output of LSTM unit.

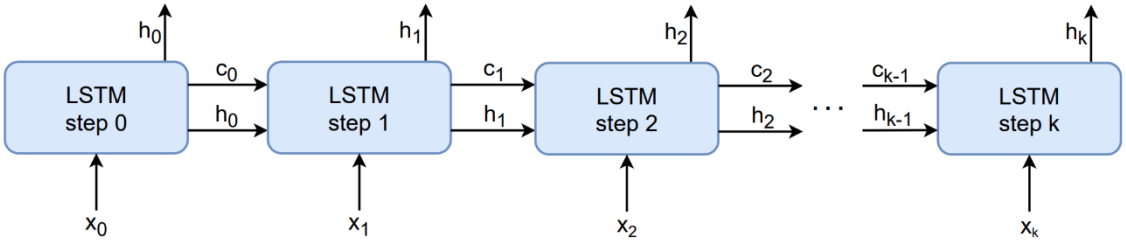


Figure 40. The repeating of LSTM units for k time steps.

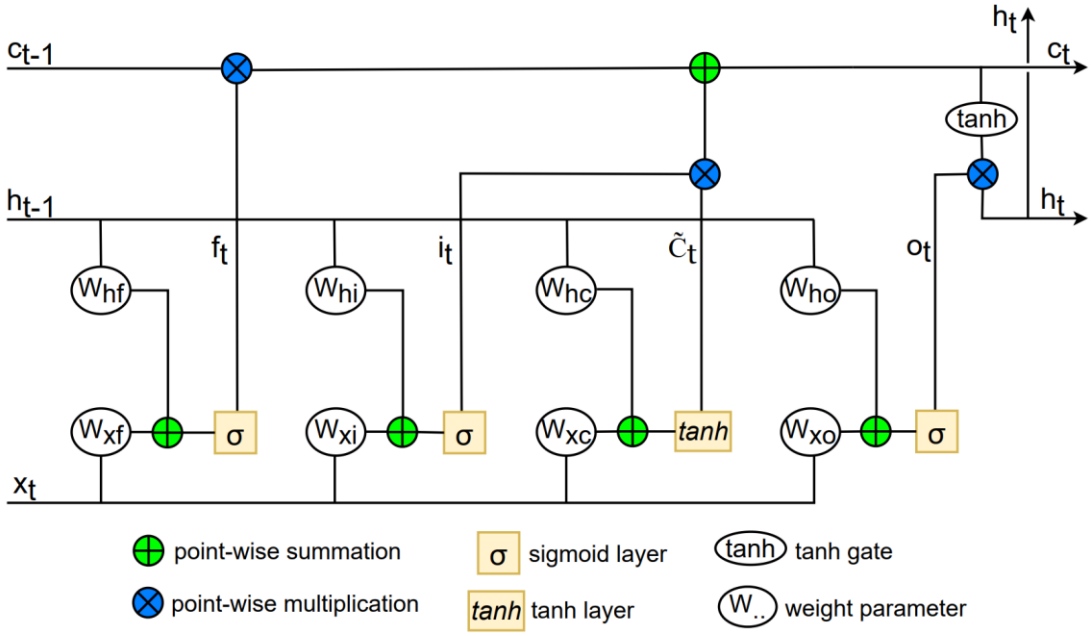


Figure 41. The architecture of LSTM network used in the present study.

The mechanism of LSTM can be described in four steps. The first step is called the forget stage, where the LSTM decides which information from its last cell state should be eliminated. This decision is implemented by a sigmoid layer with two inputs x_t and h_{t-1}. When the control signal is 0, the amount of data propagated is 0 percent. As the signal is 1, there is 100 percent of data propagated to the cell c_{t-1}. This step can be described in the equation as follows:

$$f_t = \sigma\left(W_{hf}.h_{t-1} + W_{xf}.x_t + b_f\right) \tag{32}$$

where σ is a *sigmoid* function defined as $\sigma(x) = 1/(1+e^{-x})$, W_{hf} is the weight matrix associating the hidden state h_{t-1} at the previous step $t-1$ with the forget control gate at the current state, W_{xf} is the weight matrix associating the input x_t with the forget control gate at the current state, and b_f is the bias vector.

The second step is the store stage to determine what new data will be stored in the cell state. A sigmoid layer is called again to select values to be updated, i.e. the values of the input stage are between 0 and 1, as in Eq. 32 as follows:

$$i_t = \sigma(W_{hi}.h_{t-1} + W_{xi}.x_t + b_i) \tag{32}$$

where W_{hi} is the weight matrix associating the previous output h_{t-1} with the input control gate at the present step, W_{xi} is the weight matrix associating the input x_t with the input control gate, and b_i is the bias vector. Then, a tanh layer is used to generate an array to save new information in the long-term. Note that the tanh function returns the value in range of -1 and 1, so the new information is only added to the cell state at the current time step if the returned value is positive. Eq. 33 describes this phase as follows:

$$\tilde{c}_t = tanh(W_{hc}.h_{t-1} + W_{xc}.x_t + b_c) \tag{33}$$

where *tanh* function is defined as $tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$, W_{hc} and W_{xc} are the weight matrices connected to the hidden output h_{t-1} and the input x_t, respectively, and b_c is the bias vector.

The next step is to update the old cell state into the new one. The values obtained from the second step are integrated together to add an update to the cell state. While the old state c_{t-1} and the term f_t from the first step are multiplied, then the new state is calculated as follows:

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{34}$$

The final step can be called the output stage. The information impacts on the output is determined through a sigmoid layer as shown below:

$$o_t = \sigma(W_{ho}.h_{t-1} + W_{xo}.x_t + b_o) \tag{35}$$

where W_{ho} and W_{xo} are the weight matrices connected to the hidden output h_{t-1} and the input x_t, respectively, and b_o is the bias vector. To compute the current hidden state, we use the o_t and a tanh function of the new cell state gained in the third step. This step is formulated as in Eq. 36:

$$h_t = o_t * \tanh(c_t) \tag{36}$$

All weighted and biased parameters of the LSTM network can be summarized as follows:

$$\Theta \equiv \begin{Bmatrix} W_{hf}, W_{xf}, b_f, W_{hi}, W_{xi}, b_i, \\ W_{hc}, W_{xc}, b_c, W_{ho}, W_{xo}, b_o \end{Bmatrix} \tag{37}$$

We also used a variant of long short-term memory that is a bidirectional LSTM (BiLSTM) [195], [196]. When the input series is large and the model has to understand the association across future and past information. It is necessary for us to deliver data in such a manner. In order to solve this issue, a bidirectional network was developed. In BiLSTM model, the input is provided from both the left-to-right and the right-to-left directions simultaneously. Take notice that this is not a case of backward propagation; rather, this only refers to the input that is supplied from both sides. Therefore, the issue that has to be answered is how the data from the two inputs are integrated into one output. A mechanism for combining both is known as a merge stage. The merging can be the functions of sum, multiplication, averaging and concatenation. Figure 42 describes the overview structure of the used BiLSTM network, where X_t is the input vector, Y_t is the output vector that is the combination of LSTM forward and LSTM backward nodes, the green arrows denote the forward sequence and the red arrows are for the backward sequence.

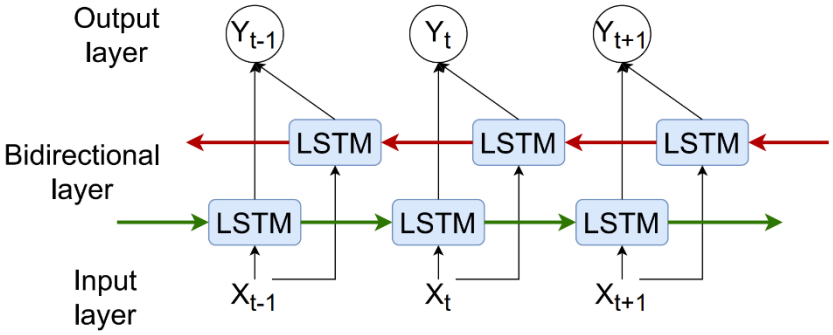


Figure 42. An architecture of the used bidirectional LSTM for pelvis soft tissue prediction.

4.2.3.2 Deep Neural Network (DNN)

To compare with LSTM architecture on the prediction of pelvis soft tissue deformation, a classical deep neural network was used [197]. The structure of this traditional network includes an input layer, layers that are concealed, and an output layer. A node in the current layer has a weighted connection to all nodes in the next layer. Further, every node in the next layer is supplied a summation that is the product of weighted parameters with the outcomes, which respect to nodes of the current layer, then that node provides output applied activation function for the summation according to the following formula:

$$o_i^{k+1} = f\left(u_i^{k+1}\right) = f\left(\sum_{j=1}^{N_k} w_{ij}^k \times o_j^k + \theta_i^{k+1}\right) \quad (38)$$

where o_i^{k+1} and u_i^{k+1} are output and input of activation function f of node i-th in the layer $k+1$, w_{ij}^k denotes the weighted parameter connected node i-th in layer $k+1$ to node j-th in layer k, and θ_i^{k+1} is a bias parameter. N_k is the number of nodes in the layer k, o_j^k is the output of activation function of node j-th in layer k.

4.2.3.3 PCA-based learning strategy

Due to the high dimensionality of the input data, we opted to apply a model reduction technique to investigate the impact of model reduction on the accuracy of a deep learning model. The primary objective of model reduction in this context is to effectively decrease the model's dimensionality by identifying and utilizing the essential '*snapshots*' that define the physical quantities present in the model. One widely used technique for model reduction is Principal Component Analysis (PCA) that employs Singular Value Decomposition (SVD) on the data covariance matrix to extract the principal components, which inherently represent the directions of maximum variance in the data. An alternative model reduction technique based on SVD is Proper Orthogonal Decomposition (POD). PCA and POD are closely linked methodologies, often utilized interchangeably, particularly in fields like fluid dynamics and structural mechanics [198], [199], [200]. However, a key distinction lies in their primary application domains. While PCA is commonly employed for general datasets encompassing various data types, POD is primarily designed for spatiotemporal data applications, where temporal information plays a significant role. PCA has previously been employed to decrease the dimensionality of a finite element simulation system [201], as well as in a deep learning approach to achieve a computer graphic simulation of cloth deformations [202]. In this study, we opted to employ PCA due to our specific focus on reducing the dimensionality of spatial data without incorporating temporal information. This decision is in consideration of the fact that the neural network is tasked with predicting the subsequent state based on the current state. The goal of PCA is to identify a small set of new variables that are unobserved, which are

derived from the existing set of variables in a manner that records the majority of the information present in the original set. The transformation to project the dataset from an original space into a new space for dimensionality reduction while maintaining information in the best way possible can be defined as below:

$$X = X_M + Pb \tag{39}$$

where X is the vector of observed variables with size of N, X_M is the mean vector of the dataset with size of N, P denotes the principal components matrix with size of $N \times n$ and b is the vector of the principal component coefficients with size of n much smaller than N. The vector b is derived from Eq. 39 as follows:

$$b = argmin[(X_M - X + Pb)^T (X_M - X + Pb)] \tag{40}$$

or

$$b = (P^T P)^{-1} P^T (X_M - X) \tag{41}$$

Thus, all the dataset of nodal positions, velocities, external forces and displacements were reduced to the optimal number of principal components before transferring to training models. This means inputs of DL models are vectors principal component coefficients of position, velocity and external force. The predicted output is a vector principal component coefficient of displacement. Then, this vector is applied back to PCA Eq. 39 to obtain predictions of nodal displacement. The flow chart of our model reduction implementation is shown in Figure 43.

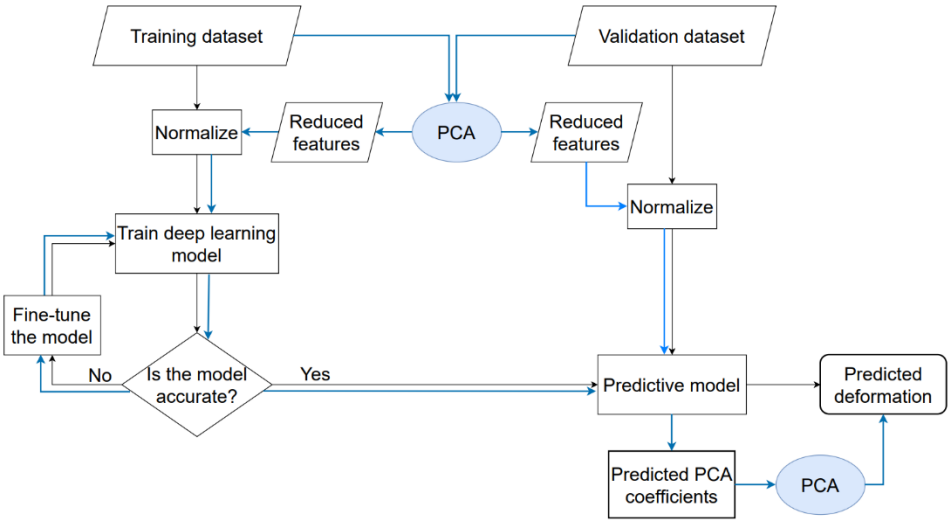


Figure 43. Flow chart of the implementation. The black line is for deep learning method, the blue line is for deep learning hybrid PCA approach.

4.2.4 Training and Evaluation

The LSTM, BiLSTM and DNN networks were trained on the raw learning database including 15396 records with the total size of input and output vectors of 19836 columns as well as on the PCA-generated database. These models were also evaluated with a testing database with 18633 records and the same number of surface nodes and columns with the training dataset. A hyperparameter tuning process on the numbers of hidden layers (from 1 to 3) and associated nodes (from 100 to 1000), batch size (from 10 to 100), learning rate (from 1×10^{-3} to 1×10^{-6}) and activation functions (Linear, Sigmoid, Tanh, ReLU) was conducted using Optuna, which is a hyperparameter optimization framework introduced by Akiba et al. [203]. In particular, dropout

technique [204] is applied on the hidden layers to prohibit neural network from overfitting, with probability of 0.005. The optimizer used to update the kernel weights is an adaptive moment estimation (Adam) that is a popular adaptive stochastic gradient descent algorithm. All computation tasks were implemented on the HPC cluster using 10 nodes with 2 CPU Intel Xeon E5 for each node of University de Lille.

Before applying the learning process, data was normalized by Eq. 42 scaling values in the range [-1,1]

$$x_t' = 2 \times \frac{x_t - x_{min}}{x_{max} - x_{min}} - 1 \tag{42}$$

where x_t' is the normalized value of x_t in the dataset X = [x_1, x_2, … , x_T], x_{min} and x_{max} are the minimum value and maximum value, respectively, in the dataset.

To evaluate the proposed model, we used the root mean square error (RMSE) (Eq. 43) and the Pearson correlation coefficient (r) as given in Eq. 44. Pearson correlation coefficient provides information about the strength and direction of the relationship between two variables [205], [206]. This coefficient is range in $-1 \leq r \leq 1$. When the direction is positive, the relationship is said to be strong when $r > 0.5$, moderate if $0.3 < r \leq 0.5$ and weak if $0 < r \leq 0.3$. If $r = 0$, the relationship between two variables is uncorrelated. With negative direction, the relationship expresses strongly when $r < -0.5$, moderate if $-0.5 \leq r < -0.3$ and weak if $-0.3 \leq r < 0$.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2} \tag{43}$$

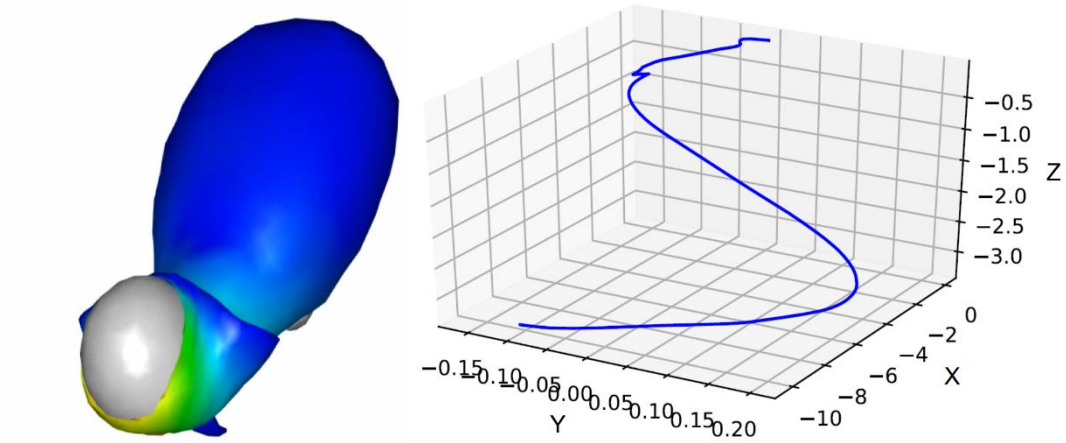
$$r = \frac{\sum_{i=1}^{N}(y_i - m_y)(\hat{y_i} - m_{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i - m_y)^2}\sqrt{\sum_{i=1}^{N}(\hat{y_i} - m_{\hat{y}})^2}} \tag{44}$$

where N is the number of samples, y denotes the actual value, $\hat{y}$ denotes the predicted value, $m_y = \frac{1}{N}\sum_{i=1}^{N} y_i$ is the mean of y, similarly, $m_{\hat{y}}$ is the mean of $\hat{y}$.

4.3 Computational results

4.3.1 Simulation-derived Learning Database

The learning database corresponds to delivery scenario under different external forces applied to push the fetal body out of the mother uterus (Figure 44). The average interval time of HyperMSM simulations was in the range of 0s to 10s with time step $\Delta t = 0.002$s. During each simulation, node position, node velocity and associated applied force were recorded and saved for further processing. Note that the maximal nodal deformations were 52.26 mm and 52.35 mm for the training database and test database, respectively.



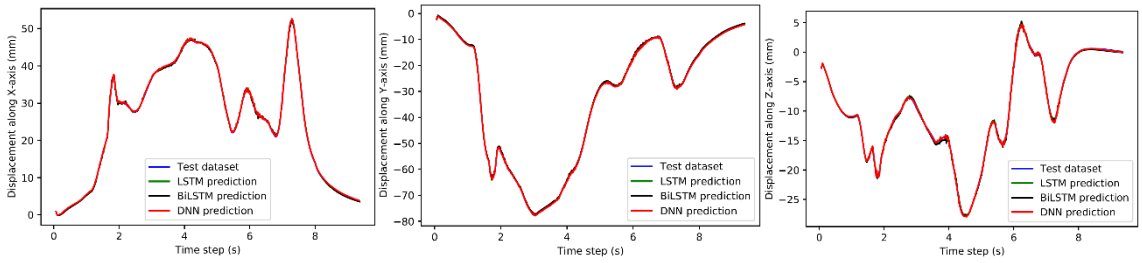
a) b)

Figure 44. 3D visualization of a simulation example in the training database (a) and associated nodal displacement of one node (b).

4.3.2 Different deep learning prediction outcomes

The evaluation of the predicted results on the training database shows a root mean square error (RMSE) of 0.173 mm, 0.168 mm and 0.139 mm for LSTM, BiLSTM and DNN models respectively. Pearson correlation coefficient (PCC) of 0.999 was obtained for all applied networks. The maximal displacement errors corresponded to 8.384 mm and 3.824 mm for LSTM and DNN models, respectively. Figure 45 displays the forecasted displacements along X-, Y- and Z-direction using different deep learning architectures with reference values of the node where the highest deformation performs in the training dataset (see Figure 46).

Regarding the evaluation on the testing database, root mean square error (RMSE) of 1.062 mm, 0.988 mm and 0.936 mm for LSTM, BiLSTM and DNN models respectively. Pearson correlation coefficient (PCC) of 0.994 was obtained for both LSTM and BiLSTM networks. A PCC of 0.996 was achieved with the DNN network. Table 3 depicts RMSE and PCC of the predicted results using LSTM models and DNN model on the training database and the testing database.



a) b) c)

Figure 45. The highest nodal displacement of the training dataset at 52.26 mm at X-axis. Plotting predicted displacements of the node using LSTM, BiLSTM models and DNN model along X- component (a), Y- component (b) and Z-component (c).

Table 3. Root mean square error (RMSE) and Pearson correlation coefficient (PCC) of predicted values using LSTM, BiLSTM models and DNN model on the training and the testing datasets.

	LSTM		BiLSTM		DNN	
	RMSE (mm)	PCC	RMSE (mm)	PCC	RMSE (mm)	PCC
Training	0.173	0.999	0.168	0.999	0.139	0.999
Test	1.062	0.994	0.988	0.994	0.936	0.996

More precisely, the best predicted case from the testing database produced mean errors of 0.103 mm ± 0.116 mm and 0.107 mm ± 0.083 mm with LSTM and DNN networks respectively (Table 4). Figure 47 compares the best predicted deformation using LSTM to HyperMSM estimation of the childbirth simulation at time step 1s, 5s, 8s and 11.186s when it obtained the maximum absolute error of 3.264 mm. By DNN model, the best prediction returned the maximum absolute error of 2.765 mm at the time step 11.092s. The visualization of this case can be seen in Figure 49.

Regarding the worst predicted case, the prediction using LSTM resulted in absolute mean error of 0.845 mm ± 1.214 mm, while the DNN-based deformation forecast has a mean error of 0.775 mm ± 1.041 mm. Figure 48 shows the LSTM prediction of this simulation in comparison with HyperMSM reference at time step 1s, 5s, 7.446s - the largest absolute displacement error here of 24.219 mm and 10s. The prediction using DNN produced the maximum absolute error of 23.293 mm at time step 7.446s as well. Note that larger errors occurred at the vaginal orifice where there is high deformation during the childbirth process. Figure 50 describes the worst case by DNN prediction over time. The rest of forecasted simulations returned mean errors from 0.266 mm ± 0.352 mm to 0.750 mm ± 0.989 mm for LSTM network and mean errors from 0.257 mm ± 0.317 mm to 0.719 mm ± 0.870 mm for DNN model. Figure 51 shows the forecasted displacements along X-, Y- and Z-direction using LSTM and DNN methods with reference values of the node where the highest deformation performs in the test dataset.

Table 4. Mean absolute error and standard deviation (SD) of predicted values using LSTM and DNN models on the testing database.

	LSTM		DNN	
Simulation	Mean (mm)	SD (mm)	Mean (mm)	SD (mm)
1	0.103	0.116	0.107	0.083
2	0.266	0.352	0.257	0.317
3	0.735	1.147	0.662	0.966
4	0.750	0.989	0.719	0.870
5	0.845	1.214	0.775	1.041
6	0.589	0.917	0.544	0.792

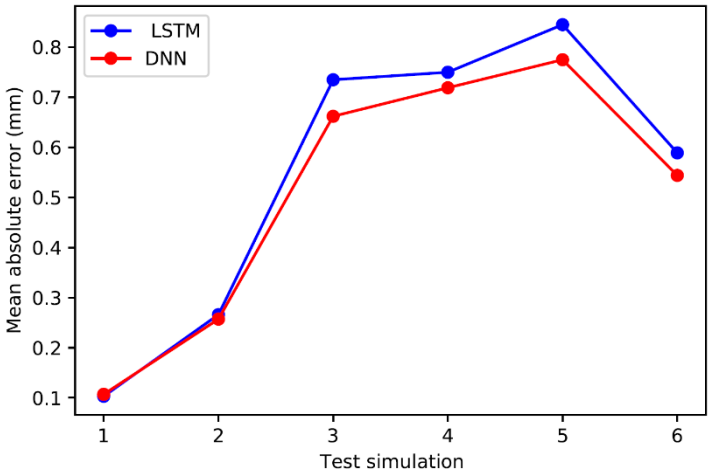


Figure 46. Comparing mean absolute error (mm) of the deformation prediction over 6 test childbirth simulations using LSTM method and DNN method.

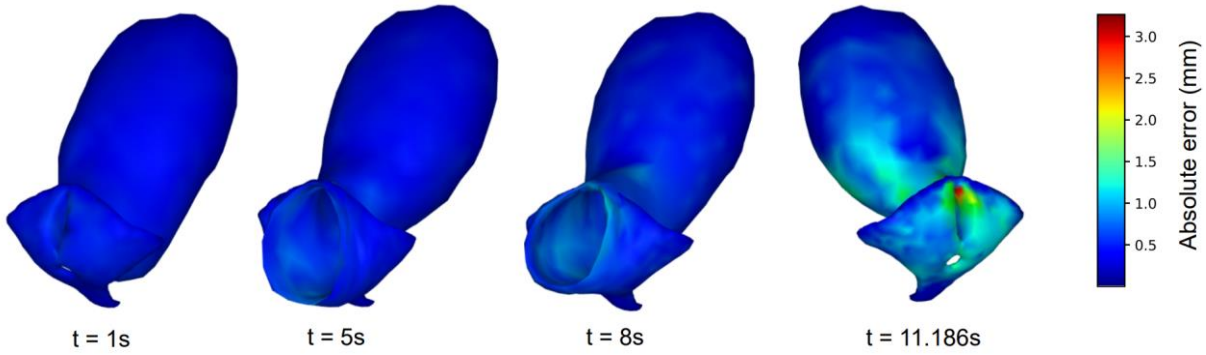


Figure 47. The best predicted simulation by LSTM over time obtained the largest absolute displacement error of 3.264 mm at time step 11.186s.

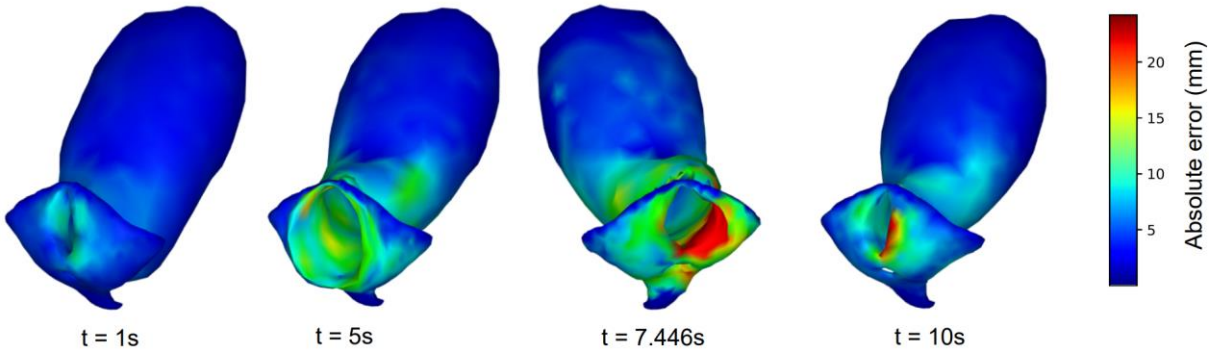


Figure 48. The worst case by LSTM prediction over time resulted in the largest absolute displacement error of 24.219 mm at time step 7.446s.

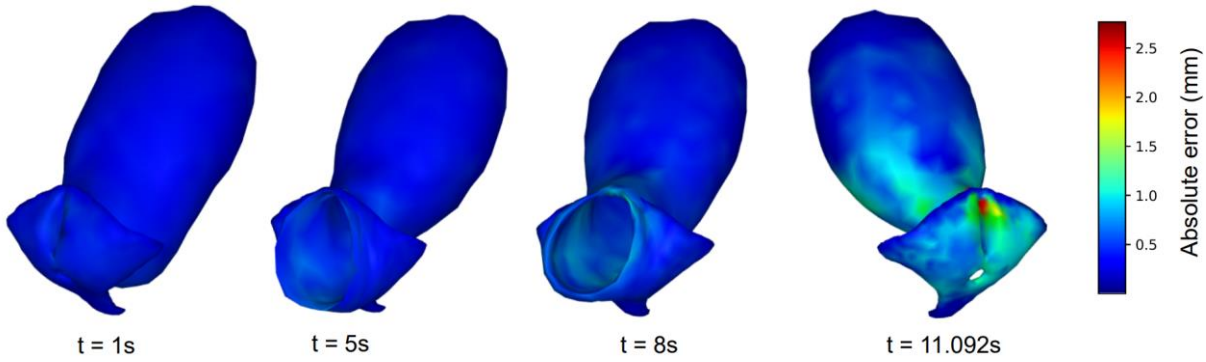


Figure 49. The best predicted simulation by DNN over time obtained the largest absolute displacement error of 2.765 mm at time step 11.092s.

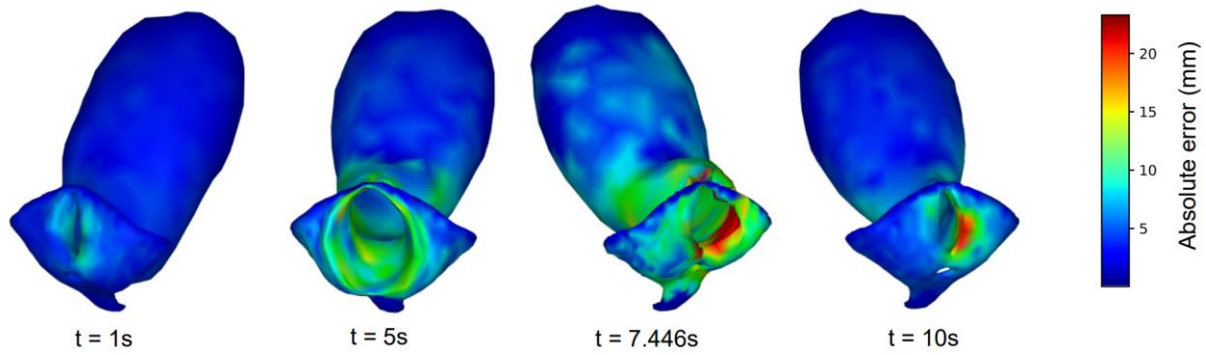
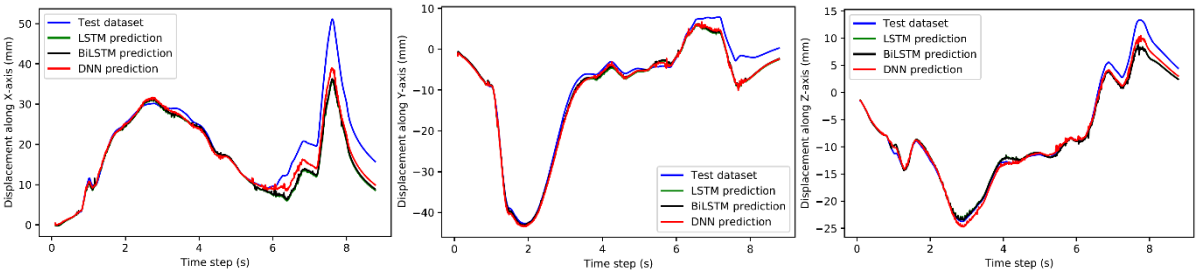


Figure 50. The worst case by DNN prediction over time resulted in the largest absolute displacement error of 23.293 mm at time step 7.446s.



a) Displacement along X-axis b) Displacement along Y-axis c) Displacement along Z-axis

Figure 51. The highest nodal displacement of the test dataset at 52.35 mm at X-axis. Plotting predicted displacements of the node using LSTM, BiLSTM models and DNN model along X-component (a), Y- component (b) and Z-component (c).

4.3.3 PCA-based learning strategy outcomes

By applying on our database, an optimal number of 11 principal components was achieved for maximizing variance of data up to 99.99% while keeping as much data as possible. The predicted results of nodal displacement by PCA-based learning strategy obtained a RMSE of 0.708 mm and mean± standard deviation of 0.438± 0.556 mm for the training dataset. The predicted results on the testing data returned RMSE from 1.535 mm to 1.547 mm and mean± SD of 0.865± 1.280 mm. Moreover, Table 5 compares the maximum absolute errors (Max.) (mm) and mean errors on the prediction validation set using three deep learning models with

three learning approaches: only training and test by deep learning (DL); training and test by DL and outputs processed by PCA; and inputs processed by PCA then by DL and outputs processed by PCA as well.

Table 5. Comparing the maximum absolute errors (Max.) (mm) and mean errors on the testing dataset using three deep learning models with three learning approaches: only training and test by deep learning (DL); training and test by DL and outputs processed by PCA; and inputs processed by PCA then by DL and outputs processed by PCA as well.

	DL		DL + PCA-derived Output		PCA-derived + DL + PCA-derived Output	
	Max.	Mean	Max.	Mean	Max.	Mean
LSTM	24.219	0.548	~25	0.869	~25	0.867
BiLSTM	23.875	0.532	~25	0.869	~25	0.868
DNN	23.293	0.511	~25	0.870	~25	0.866

4.4 Discussion

Accurate and fast prediction of biological soft tissues play an essential role in the development of the next-generation interactive decision support tools and systems. Such achievement could be of great interest in providing accurate and fast feedback (e.g. tissue displacement, deformation and stress) during different virtual interactions between tissues with and without external devices. This information is valuable for optimizing the intervention gestures or device design optimization. To reach this challenging achievement, different numerical approaches such as finite element or mass-spring model formulations have been commonly used in the biomechanics field. These approaches are usually mesh-based and then still computationally expensive leading to the limited use in the clinical routine practice. Recently, the deep learning approach has metamorphosed the biomechanical modeling and simulation with capacities to predict the biological soft tissues (e.g. skeletal muscle, brain tissue) [130], [207] deformation in a fast and accurate manner. Thus, new perspectives have been opened for innovative virtual surgical planning and decision support tools and systems. In the present study, pelvis soft tissue deformation during the delivery process of the fetal body was accurately and rapidly estimated using classical deep neural network and two recurrent neural networks (LSTM and BiLSTM). This achievement opens new avenues for the optimization of the delivery scenario to avoid complications for both fetus and mother.

Biomechanical modeling and simulation of the labor and childbirth processes are complex engineering tasks. A childbirth model is a complex structure that includes the fetal body, pelvic floor muscles, vagina and uterus covered by the pelvic bony system. The interaction between the fetal body and mother's uterus and soft tissue during different cardinal movements exhibits a nonlinear behavior. Hyperelastic laws with Neo-Hookean or Mooney-Rivlin formulations have been commonly used to model biological soft tissues in virtual interactive tools and systems due to cheaper computational cost and simpler model formulation according to more advanced laws with transversely isotropic and active behaviors [153], [154]. We showed in the present study that different deep learning models could reproduce the hyperelastic behavior of biological soft tissue in an accurate and fast manner. Our prediction accuracy level is also comparable with other studies. Recently, Liang et al. [208] presented a deep neural networks (DNN) method to predict stress distributions of the aorta. The model trained on the data of stress

distribution, which was created by finite element simulation from 729 virtual patient shapes. Each shape required the mesh of 5000 nodes and 4950 elements. The DNN configured two hidden layers with 128 nodes for each layer. The normalized mean absolute error of Von Mises stress distribution was 0.492% and peak Von Mises stress was 0.891%. Meister et al. [209] applied a deep learning method to speed up the total Lagrangian explicit dynamics model. The study simulated soft tissue deformation of a liver lobe with the mesh of 507 vertices and 1493 tetrahedra. The mean error was 0.21 mm ± 0.67 mm and the maximum displacement error was 3 mm over the time 0s to 1s. Pellicer-Valero et al. [210] predicted the real-time deformation of the liver using a feedforward neural network. The study reconstructed 154 liver geometries from Computed Tomography (CT) images, then generated 10000 simulations using finite element method. The meshes consisted of around 12000 nodes. Forces were applied on random nodes and random orientations. The study also applied the PCA method to reduce features with 27 principal components. Feedforward neural network contained a hidden layer and activation function of ReLU. The predicted displacements on a validation set of about 12.5% of the whole data obtained mean absolute error of 0.433 mm and correlation coefficient of 0.977. The use of convolutional neural network to predict the current displacement of all internal liver points when only providing the deformation of surface nodes was proposed by Pfeiffer et al. [211]. The training inputs were the surface meshes of 10000 simulations, which were generated by Elmer software. Validation of the model on 1334 samples, the estimated displacement returned a maximum error of about 25 mm and an average error of 20 mm when the whole liver's surface was observable. Madani et al. [212] estimated stress of arterial wall in atherosclerosis using finite element method (FEM)-based neural networks. From FEM simulations, parameters of arterial geometry and pressure were extracted as inputs. The network trained on 9737 simulations and test on 2435 samples. The configuration of the model included 3 hidden layers, an activation function of ReLU and Adam optimizer. The prediction of maximum von Mises stress obtained a mean squared error of 9.86% and a mean absolute error for polar distance of 0.223 mm. Finally, Mendizabal et al. [213] applied U-Net to predict lesion deformation in breast biopsy. Hexahedral meshes were generated by FEM for training data input. The proposed U-Net was trained on 800 meshes and tested on 200 samples. The network set up 100000 epochs and Adam optimizer. The predicted displacement lesion compared with real displacement extracted from ultrasound images showed a mean norm error from 2.124 mm to 6.194 mm over 10 tumors. Furthermore, Joldes et al. [214] presented a meshless method for computing soft tissue deformation that was meshless total Lagrangian explicit dynamics including a modified moving least squares method. The method was verified on the unconstrained compressed cube with a length of edge 100 mm and Neo-Hookean material. Compared to analytical solutions, the method obtained a normalized root mean square error (NRMSE) of 3.08×10^{-3} for 1014 integration points. Verifying on a sample of sheep brain in 3D, the method acquired NRMSE 7.74×10^{-3} in x-direction, 5.57×10^{-3} in y-direction and 2.03e-2 in z-direction, comparing with ABAQUS solution. A finite element method based on absolute nodal coordinate formulation to model soft tissue in 3D was introduced by Obrezkov et al. [215]. Applying the method to a cylinder with radius of 0.1 m, length of 1 m and material of Neo-Hookean, its deformation was 0.06 m when the load was 5000N. In the experiment of rectangle cross section beam with the width and height of 0.1 m, length of 1 m and also Neo-Hookean material, the displacement was 0.22 m under the load of 5000N. Our obtained results showed a relative root mean square error less than 3% and very good Pearson correlation coefficients demonstrating the strong similarity between predicted outputs and ground truth data. In particular, the computational time for training a LSTM epoch took 5 minutes, so 1000 epochs took more than 83 hours running on the HPC cluster of Université de Lille. Meanwhile, it took 3 minutes to train a DNN epoch and 250 hours to finish 5000 epochs. In general, the prediction just consumed a few seconds running on a personal computer. Thus, our proposed approach could be an alternative solution to replace

finite element or Mass-Spring model formulation in which mesh configurations are compulsory toward a meshless deep learning model for real-time soft tissue deformation estimation.

The stochastic nature of the deep learning has been a well-known characteristic in the prediction of signals with nonlinear and high-frequency oscillations. To avoid unreliable prediction outcomes, optimal hyperparameters should be selected. A rigorous hyperparameter tuning process was realized in the present study to select the optimal set of model parameters for each deep learning architecture. Moreover, repeated runs were performed to get stable outcomes. Generally, possessing an ample dataset allows us to develop a more precise predictive model due to the increased data available for training the computer. However, dealing with huge datasets comes with its own set of disadvantages. Large datasets with high dimensionality give rise to the most prevalent issue that is overfitting. Overfitting limits the ability to make generalizations beyond the scope of the training data. Consequently, it is advisable to utilize dimensionality reduction methods to decrease the number of features in the dataset. In this study, we use the principal component analysis (PCA), which enhances interpretability, minimizes information loss, assists in identifying crucial attributes and aids in identifying linear combinations of diverse sequences. However, a little effect was noted for the accuracy level while the computational cost is largely cheaper. In fact, the computational time for PCA-derived learning takes only 16 minutes.

To obtain accurate prediction with deep learning, reliable and accurate learning database should be used. In the biomechanics field, due to the impossible acquisition of soft tissue displacement and deformation in a non-invasive and *in vivo* conditions, numerical simulations have been commonly used for database generation [216]. Finite element simulations have been usually used. In the present study, our learning database was generated using our recently developed HyperMSM formation, which has a cheaper computational cost and more flexible simulation configuration capacities. Note that our novel formulation was evaluated using the finite element solution and obtained results are very comparable for hyperelastic law with Neo-Hookean formulation [159].

One of the important limitations of the present study remains in the number of delivery scenarios. Thus, only the normal delivery process of the fetal body was considered. Thus, further investigations should be performed with more complex delivery scenarios (e.g. forceps-assisted delivery or complicated delivery scenarios) toward a complete prediction of pelvis soft tissues during the childbirth process. Another limitation relates to the simplifications adopted in our childbirth model. This encompasses the geometric representation, the fetal descent model, and the node-to-triangle contact model employed to handle interactions between the fetal body and pelvis soft tissue [159]. While these simplifications are deemed suitable for the present study and adequately serve the established objectives (predict the soft-tissue deformations using machine-learning model), future studies would benefit from an anatomically accurate model equipped with an advanced contact model and physiologically simulated fetal cardinal movements to attain a heightened level of realism. Another limitation relates to the use of a simple node-to-triangle contact model for managing the interaction between the fetal body and pelvis soft tissue [159]. More complex contact model is needed in the future for a more realistic interaction. Another limitation deals with the use of recurrent neural network with our problem formulation. In fact, the long-short term memory (LSTM) networks give a similar accuracy level in comparison with classical deep neural network in our present study. Normally, this specific network architecture with memory components provide appropriate function approximations to describe the non-linear mappings between the input time-series data and the desired time-series outcomes. One possible reason remains in our

problem formulation. In fact, the prediction of the next time step information based on only the previous one is not favorable with the recurrent neural network philosophy. Thus, further studies should be investigated to propose a new model prediction formulation to take the advantage of the LSTM network into consideration. Furthermore, physics-informed neural networks will be investigated to add direct physical properties of the problem formulation itself to guide the training process leading to improve the prediction accuracy. In particular, the use of our novel mass-spring system modeling (i.e. HyperMSM) approach makes the direct comparison with other state-of-the-art outcomes using deep learning for soft tissue prediction unfeasible. Thus, traditional finite element modeling approach will be used to generate the training data to enhance the finding of the present methodology. In addition, non-learning neural network methodologies for modeling soft tissue deformation should be also potential solutions to deal with the lack of representative and realistic training data [217], [218]. Finally, the present study focused on the complexity of the coupling between finite element modeling outcomes and different deep learning techniques for solving real-time soft tissue deformation. Thus, the state-of-the art deep learning models were used. Further improvements on the used methods will be considered in a future study.

Chapter 5

Physics-Informed Neural Ordinary Differential Equations and Mass-Spring System Modeling Framework for Predicting the Real-Time Soft-Tissue Deformations

Expanding upon the research conducted in Chapter 4, this chapter endeavors to advance childbirth simulation by simulating soft tissue deformations in real-time. Traditional numerical methods, reliant on mesh configurations, are fraught with high processing costs and instability in convergence. To address these challenges, we turn to physics-informed neural networks (PINN), recognized for their efficacy in solving differential equation problems. Our approach integrates PINN and neural ordinary differential equations (NeuralODE) with the mass-spring system (MSS) modeling framework to develop a predictive model for soft tissue deformation. We establish the theoretical underpinnings of our methodology using PINN, NeuralODE, and MSS formulations. Subsequently, we assess the efficacy of our novel approach through simulations on 2D and 3D mass-spring systems, as well as a 3D cantilever beam. Finally, we apply our model to simulate and evaluate the deformation of a 3D uterus model under gravity loading, marking a significant step forward in the development of realistic childbirth simulations.

5.1 Introduction

Machine learning (ML) and deep learning (DL) have been emerging as pivotal methods for propelling scientific study and computational advancements across various domains, including solid mechanics [168], [219], biomechanics [129], [130] and computer vision [132], [133]. ML and DL were initially proposed in computational science to overcome inefficient data modeling techniques that hindered rapidly engaging with complicated data. These methodologies have revolutionary possibilities for exploring large design environments, identifying multidimensional relationships, and addressing complex difficulties. Nevertheless, traditional ML and DL techniques struggle to derive meaningful insights and expertise from intricate multi-dimensional data. While they can efficiently map observations or computational data, their predictions might lack physical rationality or credibility, leading to subpar generalization.

The mechanical functions of the human body depend essentially on the movements of soft tissues. When these tissues interact to other tissues and external equipment, they display commonly non-linear characteristics and behaviors. Numerical models have been developed to assess and predict soft tissue deformations. Ballit and Dao [159] introduced a hyperelastic mass-spring model, called HyperMSM, for simulating deformations of Neo-Hookean material. The model supported tetrahedral and hexahedral meshes and was evaluated for its performance on skeletal muscles and residual limb deformation. Santesteban et al. [94] offered a data-driven approach to represent soft-tissue dynamics while considering both body shape and motion. The

model unraveled the conventional pose representation by eliminating features specific to individual subjects. Nguyen-Le et al. [131] suggested the use of deep neural networks and long short-term memory to forecast the displacement of the uterus during the second stage of labor simulation. Various learning strategies, including those with and without data dimension reduction mechanism, were applied. Despite advancements, challenges persist in terms of precision, computational cost and speed when simulating in real-time configuration.

Recently, there has been a demonstrated benefit in combining differential equations with machine learning. Thus, physics-informed neural networks (PINN) were first introduced by Raissi et al. [107]. They execute supervised learning while adhering to prescribed physical laws, which are presented in the form of differential equations. The PINN methods adhere to the rules of physics by combining a weakly enforced loss function that includes the residuals from physics equations and initial/boundary conditions. To compute derivatives of the neural network outputs concerning inputs such as spatiotemporal and model parameters, the automated differentiation [98] is employed. The networks minimize the loss function to approach solutions approximately. There have been numerous studies employing PINNs to solve partial differential equations. For example, Arnold and King [220] provided a state-space modeling using the PINN. The model eliminated the need for numerical solution techniques in state propagation. Instead, each time step relied on assessing a moderately sized neural network, which approximated the solution of dynamic systems illustrated by the Kalman filter and Burger's equation. Schaffer et al. [221] presented the use of PINN to examine the complete three-dimensional static micromagnetic equations approach for the continuous magnetization configuration. The model was trained to minimize the Gibbs free energy using a low-parametric network. Additionally, the study used PINN to address the significant and computational cost stray field problem. In particular, the practicality of identifying nonlinear structures has been investigated by employing neural ordinary differential equations (NeuralODE) [99] as a means of approximating governing equations. NeuralODEs, a category of deep neural network models, offer a framework that connects neural networks - commonly associated with data-driven approaches - with differential equations, which are rooted in physics. These models have demonstrated their ability to learn the inherent governing dynamics from observed data of a dynamic system. Luo et al. [222], for example, constructed predictive control models using NeuralODE. A range of data processing approaches were used to ensure closed-loop stability and address resilience to noise. The study examined the work of the proposed control models when subjected to both Gaussian and non-Gaussian noises. Roehrl et al. [223] combined the physics-informed neural network and NeuralODE to simulate a forward model of an inverted pendulum on a cart. The results showed that it was capable of learning the non-conservative forces in a physically actual system with a significant degree of uncertainty.

Numerous open-source software libraries tailored for physics-informed neural networks have been created, which is fueling the discipline's explosive growth. Xu and Darve [224] introduced the ADCME library to lean spatially-varying physical fields. In order to facilitate the execution of physics-informed learning, ADCME operates as a wrapper, converting low-level functions from different libraries into comparatively high-level operations. Users remain responsible for carrying out all necessary steps to solve the issue. Hennigh et al. [225] developed SimNet for Nvidia GPU to simulate large-scale problems. The toolkit considers not only the system's geometry but also an explicit parameterized space of the input geometry. Another wrapper tool like ADCME is SciANN introduced by Haghighat and Juanes [226]. Certain software, like DeepXDE [227], can function as a solver. This library is capable of solving ordinary differential

equations (ODEs), partial differential equations (PDEs), integro-differential equations, and fractional PDEs. Additionally, it accommodates a wide range of domain geometries. An extensive collection of software packages centered around physics-informed learning is found in SciML [228], [229]. This ecosystem includes numerous libraries designed for addressing various types of equations such as ODEs, stochastic differential equations, differential-algebraic equations, and hybrid differential equations, encompassing multi-scale models and combinations with agent-based simulations. Each of these libraries employs automatic differentiation, which is available in other software tools like TensorFlow.

In this work, we developed and evaluated a suitable predictive model of the soft tissue deformation based on PINN approach and physics-informed hybrid neural ordinary differential equations (PINODE) coupled with the mass-spring system (MSS) modeling framework. Our proposed approach was evaluated on 2D and 3D mass-spring systems as well as a 3D cantilever beam. Then, its application on the soft tissue deformation of the 3D uterus organ was performed and evaluated.

5.2 Methods

5.2.1 Theoretical basis: governing equation of the mass-spring model (MSM) for soft tissue deformation

In this study, we adopted the mass-spring model (MSM) approach to describe the soft tissue deformation [159]. This approach is frequently driven by its uncomplicated expression of constitutive equations and its ability to execute real-time calculations with adequate physical correctness.

Assuming that a dynamical system is constructed from *s* springs with *m* mass points in the *r*-dimension (*r*=2,3). In this study, we solve the following equation of motion:

$$M\ddot{x} + D\dot{x} + Kx = F \tag{45}$$

where M is the mass matrix, K is the stiffness matrix, F denotes the external force vector and D is the damping matrix given by the following equation:

$$D = \alpha_1 M + \alpha_2 K \tag{46}$$

with $\alpha_1$and α_2 are two constants.

According to Hooke's law, the potential energy in a spring is expressed as follows:

$$E = \frac{1}{2} k(\|p_1 - p_2\| - d)^2 \tag{47}$$

where k is a stiffness of the spring, $p_1, p_2 \in R^r$ are two spring endpoints. d is a vector that characterizes the equilibrium state of a spring, represented by a vector with a norm equal to the rest length. It is adjusted based on changes in nodal positions.

By summing the presentations of all springs in a system and applying matrix calculus, the total spring potential energy is obtained as follows:

$$E_s(\mathrm{x}) = \frac{1}{2}\sum_{i=1}^{s} k_i \left\| p_{i_1} - p_{i_2} - d_i \right\|^2 = \frac{1}{2}\mathrm{x}^T L\mathrm{x} - \mathrm{x}^T Jd \tag{48}$$

where L is a Laplacian matrix based on the stiffness values and the connections among nodes, J is a matrix that outlines the allocation of nodes to the springs. These two matrices are calculated as follows:

$$L = \left(\sum_{i=1}^{s} k_i A_i A_i^T \right) \otimes I_r \tag{49}$$

$$J = \left(\sum_{i=1}^{s} k_i A_i S_i^T \right) \otimes I_r \tag{50}$$

where $A_i \in R^m$ stands for the incidence vector corresponding to the *i-th* spring given by $A_{i,i_1} = 1, A_{i,i_2} = -1$, and zero otherwise. $S_i \in R^s$ represents the *i*-th spring indicator, $S_{i,j} = \delta_{i,j}$. The matrix I_r denotes the identity matrix and operator $\otimes$ is the Kronecker product.

The internal spring forces, which arise from the restoring tendency of the springs to return to equilibrium positions, are given as follows:

$$F_s(\mathrm{x}) = -\nabla E_s(\mathrm{x}) = -L\mathrm{x} + Jd \tag{51}$$

By combining the Eq. 45 and 51, we have the equation to describe the mass-spring systems in 2D and 3D as follows:

$$M\ddot{x} + D\dot{x} + Lx - Jd = F \tag{52}$$

5.2.2 Physics-informed neural network (PINN) coupled to the MSM model for soft tissue deformations

The approach of PINN differs significantly from standard supervised machine learning, as it does not solely depend on data; rather, it leverages the physical properties of the differential equations to direct the training process. The inclusion of known data into the loss function based on physics enhances the training speeds. PINNs exploit two fundamental features of neural networks. First, neural networks serve as universal function approximators. As a result, if a neural network is deep and powerful enough, it can estimate any function, including the solution to the differential equations. Second, computing derivatives of outputs of a neural network concerning its inputs, as well as model parameters during backpropagation, is straightforward through automatic differentiation.

Suppose the targeted solution of the ordinary differential equation Eq. 52 is $u(t)$. We seek an approximated solution using a neural network, represented as $\hat{u}(t;\theta)$, which is $\hat{u}(t;\theta) \approx u(t)$. Here, t denotes different time values, and θ is a set of parameters in the neural network. The optimization problem is based on the two loss functions as follows:

$$L = w_{data} L_{data} + w_{ODE} L_{ODE} \tag{53}$$

where w_{data} and w_{ODE} are parameters that need to be determined. L_{data} is defined as follows:

$$L_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} (\hat{u}(t_i) - u_i)^2 \tag{54}$$

where N_{data} is a dataset of known solution at different times including the initial and boundary conditions. The term L_{ODE} is defined as follows:

$$L_{ODE} = \frac{1}{N_{ODE}} \sum_{j=1}^{N_{ODE}} \left(M \frac{d^2 \hat{u}(t_j)}{dt} + D \frac{d\hat{u}(t_j)}{dt} + L\hat{u}(t_i) - J\mathrm{d}\left(\hat{u}(t_j)\right) - F \right)^2 \tag{55}$$

where N_{ODE} is a set of collocation points in a domain. Figure 52 provides an overview of the structure of the vanilla physics-informed neural network.

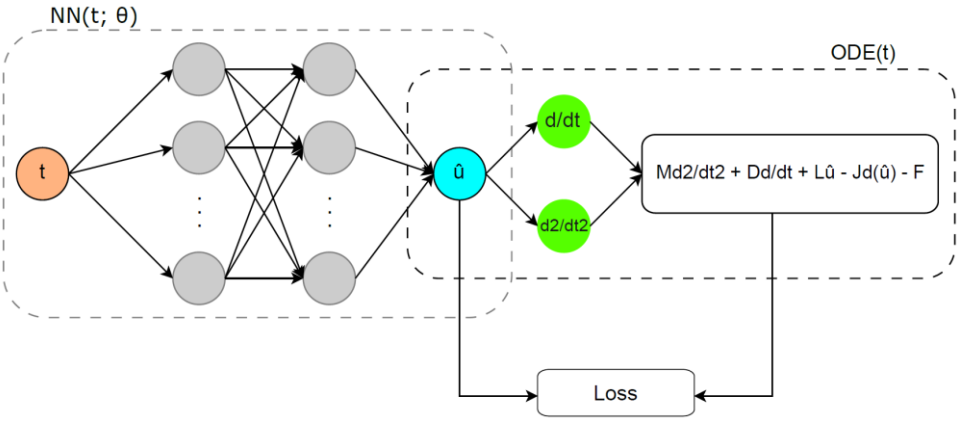


Figure 52. Overview of the architecture of the physics-informed neural network.

5.2.3 Neural ordinary differential equations (NeuralODE) coupled to the MSM model for soft tissue deformations

The fundamental concept of NeuralODE relates to the consideration of the hidden states of a neural network as outcomes of a system of ordinary differential equations [99]. In conventional neural networks, the learning procedure is based on stacking neural layers and applying a series of parameterized transformations to the input data. In contrast, NeuralODE approaches the problem from a continuous-time perspective using ordinary differential equations to define the evolution of the hidden layers of a neural network over time. A hidden state in a residual network, for example, enables the transition from layer t to layer t+1 using a neural network f as follows:

$$h_{t+1} = h_t + f(h_t, \theta_t) \tag{56}$$

where θ_t denotes a set of parameters at layer t.

Rather than employing discrete layers connecting inputs and outputs, we consider the hidden layers as continuous by parameterizing them through the limit where a time step Δt approaches zero as follows:

$$\frac{h_{t+1} - h_t}{\Delta t} = \frac{dh(t)}{dt} = f(h(t), t, \theta) \tag{57}$$

where $h(t)$ is the value of the hidden state at some time t with input layer $h(t_0) = h_0$, $f(.)$ governs the evolution of the state concerning time, and θ presents all the parameters of the network.

We can solve the Eq. 57 by deriving the integration as follows:

$$h(t) = \int f(h(t), t, \theta)dt \tag{58}$$

With the initial condition $h(t_0) = h_0$, the output $h(t_p)$ after p time steps can be derived from Eq. 58 as follows:

$$h(t_p) = h(t_0) + \int_{t_0}^{t_p} f(h(t), t, \theta)dt \tag{59}$$

We can solve the Eq. 59 by numerical methods through a black-box ODE solver as follows:

$$h(t_p) = ODESolve(h(t_0), t_0, t_p, \theta, f) \tag{60}$$

Then, it is essential to evaluate a loss function by backpropagating through the ODE solver to determine the parameters θ of the neural network as follows:

$$L\left(h(t_p)\right) = L\left(ODESolve(h(t_0), t_0, t_p, \theta, f)\right) \tag{61}$$

The gradients $\frac{\partial L}{\partial \theta}$ can be computed by an adjoint sensitivity method, which can help to lower memory costs. To further optimize L, we apply optimization methods such as stochastic gradient descent (SGD), adaptive moment estimation (Adam). The time derivative $\frac{dh(t)}{dt}$ is approximated by the neural network $f(.)$, which is learned during the training process.

5.2.4 Physics-informed neural ordinary differential equations (PINODE) coupled to the MSM model for soft tissue deformations

We explore the combination of physics-informed neural networks and neural ordinary differential equations to solve our dynamic system related to the soft tissue deformation. It is noted that the structure of NeuralODE conforms to the standard dynamic system form when considering the hidden state $h(t) = [x(t)\ \dot{x}(t)]^T$, where $x(t)$ denotes the displacement vector and $\dot{x}(t)$ denotes the velocity vector. We know that the majority of practical structural dynamics issues involve forced vibration rather than being limited to initial value problems. We adjust governing equations to following equation with the aim of including domain knowledge to the design of NeuralODE:

$$\frac{dh(t)}{dt} = f_{phy}\big(h(t), t, F(t)\big) + NN(h(t), t, \theta) \tag{62}$$

where $f_{phy}(.)$ is a physics-informed term guided by potential prior domain knowledge, $NN(.)$ represents a feed-forward neural network. $f(.)$ in Eq. 57 has been divided into these two terms. $F(t)$ represents a vector of recognized external forces exerting on the system.

Inserting Eq. 52 into Eq. 62, we have the state-space form as follows:

$$\frac{dh(t)}{dt} = \begin{bmatrix} \boldsymbol{O^{(1)}} & \boldsymbol{I} \\ -M^{-1}L & -M^{-1}D \end{bmatrix} h(t) + \begin{bmatrix} \boldsymbol{O^{(2)}} \\ M^{-1}J \end{bmatrix} d + \begin{bmatrix} \boldsymbol{O^{(3)}} \\ M^{-1}F \end{bmatrix} + \begin{bmatrix} \boldsymbol{O^{(4)}} \\ NN\big(h(t)\big) \end{bmatrix} \tag{63}$$

where $O^{(.)}$ stand for zeros vectors with different sizes, $O^{(1)} \in R^{dof \times dof}$ (dof: degree of freedom), $O^{(2)} \in R^{dof \times s}$, $O^{(3)} \in R^{dof \times 1}$ and $O^{(4)} \in R^{dof \times 1}$. The identity matrix $I \in R^{dof \times dof}$.

5.2.5 Numerical examples and uterus soft tissue deformation application

5.2.5.1 2D/3D mass-spring systems and 3D cantilever beam

a) 2D mass-spring system

The 2D mass-spring model system includes one spring and one mass. The spring was fixed at a point as in Figure 53, so the number of degrees of freedom is 2. The stiffness of the spring is 10 N/m, the weight of the mass is 1 kg, and two damping coefficients are $\alpha_1 = 0, \alpha_2 = 0.001$. We simulated the motion of the system using PINN and PINODE methods with the initial positions $x(t_0 = 0) = [1, 0]$, initial velocities $\dot{x}(t_0 = 0) = [0, 2]$, and the external forces are zero. We used the Runge-Kutta method to obtain data solution as the ground truth. The total time simulation was in 10 s, we split the first 5 s for training neural networks, then the obtained models extrapolated the motion of system in the time interval [5, 10] s. The time step size in this example is 0.05, so there are 100 data points in the training set.

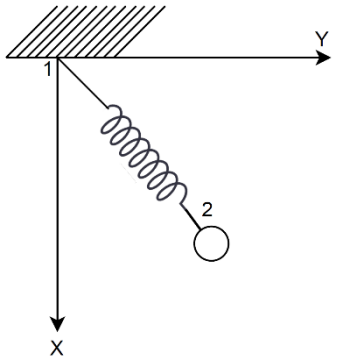


Figure 53. A mass-spring system in 2D including a spring and a mass.

To implement the PINN model, the number of hidden layers is 3 with 60 nodes per each layer. The learning rate is set at 0.001, the number of epochs is 250000. The activation function is *sigmoid*. For the PINODE, we trained the model through 20000 iterations. The learning rate starts at 0.05 and reduces 5 times after each 5000 iterations. The hidden layer consists of 1 layer with 10 nodes, and the activation function is *tanh*. We used the Google Colab platform to run implementations of the physics-informed neural models.

b) 3D mass-spring system

Given a 3-dimensional mass-spring system as illustrated in Figure 54. Three masses are distributed within a tetrahedral structure, connected by 6 edge springs. The system exhibits 9 degrees of freedom. Each mass has a weight of 1 kg. The stiffness of each spring is 10 N/m. The damping constants are $\alpha_1 = 0.01, \alpha_2 = 0.01$. We applied constant forces of 10 N on all three dimensions of the nodes. Utilizing a time step size of 0.05 s and generating a dataset comprising 200 data points for the training phase in 10 s. Following this training period, the models were tested for ability to extrapolate by predicting positions from 10 s to 20 s. We employed the Runge-Kutta method to derive the solution data, which serves as the ground truth.

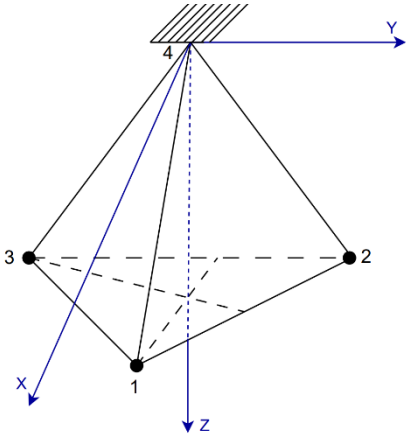


Figure 54. Tetrahedral structure of the 3D mass-spring system.

In implementing the PINN, a model architecture with 3 hidden layers, each comprising 50 nodes, was employed. The neural network was trained over 500000 epochs. The learning rate is 10^{-6}. The activation function chosen for the network is the *sigmoid* function. To the PINODE framework, we used a single hidden layer containing 30 nodes. The *tanh* function was chosen as the activation function. The training process loops over 40000 iterations with a learning rate schedule. The initial learning rate is set at 0.05 and is reduced by a factor of 5 after every 10000 epochs.

c) 3D cantilever beam system

In this example, we studied the deformation of a cantilever beam under the influence of gravity, as illustrated in Figure 55. The beam was constructed from a Neo-Hookean material characterized by a Young's modulus of $E = 0.1$ MPa and a Poisson's ratio of $\nu = 0.4$. The dimensions of the beam were 0.04 m in width, 0.04 m in height and 0.1 m in length. The structural mesh included 430 tetrahedral elements interconnected by 136 nodes. We fixed 17 nodes at the right-hand side. With 119 active nodes, the system exhibits a total of 357 degrees of freedom.

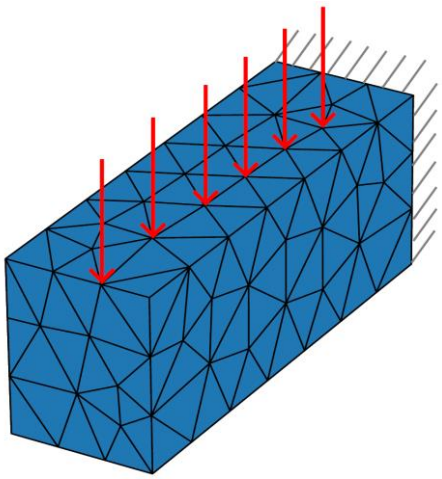


Figure 55. Cantilever beam under the gravity force.

We simulated the beam deformation in 5 s. The first 2 s was dedicated to training the PINODE model, followed by extrapolation for subsequent 3 s. The model consists of one hidden layer with 10 nodes. The *tanh* activation function was employed. We set the learning rate of 0.05 and

we trained the model through 3000 epochs. Unfortunately, implementing the PINN method in Python exceeds the memory resource available in Google Colab. Consequently, we compared the performance of PINODE method with the ground truth data, which was generated using the ODE backward differentiation formula solver in the Sundials package with Julia code. The length of the training set was 1000 data points.

5.2.5.2 3D uterus soft tissue model

Our proposed approach was applied to the biomechanical analysis of the soft tissue of uterus, which is developed by using the Neo-Hookean material with Young's modulus of 0.06 MPa and a Poisson's ratio of 0.45. In this simulation, we examined the displacement of the uterus under the gravity loading. The structural representation consists of 1389 nodes with 1347 nodes marked as active and 42 nodes fixed at the left-hand side. We had a mesh of 4077 tetrahedral elements. The simulation was characterized by 4041 degrees of freedom. Figure 56 depicts the initial state of the uterus.

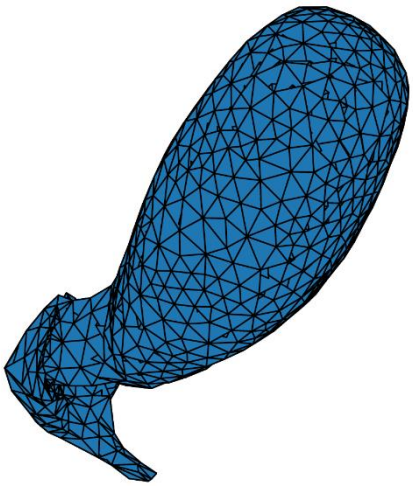


Figure 56. Initial state of the uterus soft tissue model.

In the simulation, the time of 2 s interval was divided into two phases. The first 1 s is allocated for training, and the subsequent 1 s involves extrapolation. The time step size is 0.002 throughout the simulation. The training dataset included 500 points. We applied the ODE backward differentiation formula solver to generate the data for reference. The configuration of the PINODE model consists of one hidden layer with 10 nodes. The training process involves 4000 iterations with a learning rate of 0.05 for the initial 2000 iterations and a reduced learning rate of 0.01 for the remaining epochs. The activation function employed was the *tanh* function. The implementation of PINN faced memory limitations on Google Colab when handling simulations of hundreds of degrees of freedom in the dynamic system due to exceeding the available memory resources.

Note that, for the implementation of PINN in Python, we employed the DeepXDE library, which is an efficient tool for the PINN algorithm to solve forward and inverse problems in partial differential equations. For PINODE, we turned to Julia, a high-performance programming language, utilizing libraries such as DiffEqFlux, DifferentialEquqtions from SciML. Finally, we applied the optimizer of adaptive moment estimation (Adam) in all experiments.

5.2.5.3 Performance metrics

We implemented three different performance metrics (root mean square error, L^2 norm relative error, and Pearson correlation coefficient (PCC)) to evaluate the outcomes of the performed numerical examples and the uterus model. The root mean square error is a commonly used metric to measure the average magnitude of the errors between predicted values and ground truths. The RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{64}$$

where n is the number of observations or data points, x_i is the i-th ground truth data point and $\hat{x}_i$ is the i-th predicted data point.

The L^2 norm relative error is a measure of the difference between two vectors in terms of their Euclidean norms. The following equation defines the L^2 norm relative error:

$$L^2\ relative\ error = \frac{\left\|X - \hat{X}\right\|_2}{\|X\|_2} \tag{65}$$

where X denotes the ground truth, $\hat{X}$ denotes the predicted vector and $\|\cdot\|_2$ is given as:

$$\|X\|_2 = \sqrt{\sum_{i=1}^{n} {x_i}^2} \tag{66}$$

In addition, the most prevalent method for quantifying a linear correlation is the Pearson correlation coefficient, which is a numerical value ranging from -1 to 1. This coefficient gauges both the magnitude and direction of the association between two variables. The formula for calculating the Pearson correlation coefficient is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - m_X)(\hat{x_i} - m_{\hat{X}})}{\sqrt{\sum_{i=1}^{n}(x_i - m_X)^2}\sqrt{\sum_{i=1}^{n}(\hat{x_i} - m_{\hat{X}})^2}} \tag{67}$$

where $m_X = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean of X and $m_{\hat{X}}$ is the mean of $\hat{X}$.

5.3 Computational results

5.3.1 2D mass-spring systems

For the predictions from the training set, the PINN model returned the RMSE of 1.034 $\times 10^{-3}$ m and L^2 norm relative error of 0.122%, whereas the PINODE obtained higher errors with 8.549 $\times 10^{-3}$ m for the RMSE and 0.982% for L^2 norm relative error. However, the capacity of extrapolation of PINODE performed much better than PINN. The RMSE and L^2 norm relative errors of the predicted results were 0.117 m and 13.718% using PINODE, and 1.491 m and about 174% using PINN, respectively. The PCC of -0.139 demonstrated that the prediction by PINN model was a weak negative correlation. All the errors in the predictions are given in Table 6. In Figure 57 and Figure 58, we can see that extrapolated curves used PINODE fit well to the ground truth, but PINN extrapolated the simple ones. Figure 59 shows that the RMSE when

predicting positions of the mass in the time from 5 s to 10 s using PINN were significantly higher than using the PINODE.

Table 6. The performances of the predicted trajectories of the 2D mass-spring dynamic system from the training during 5s and the extrapolation from 5s to 10s using the PINODE and PINN models. The terms of performance metrics are the root mean square error (RMSE), L^2 norm relative error and Pearson correlation coefficient (PCC).

	PINODE			PINN		
	RMSE (m)	L^2 (%)	PCC	RMSE (m)	L^2 (%)	PCC
Training	8.549×10^{-3}	0.982	0.999	1.034×10^{-3}	0.122	0.999
Extrapolation	0.117	13.718	0.990	1.491	174.099	-0.139

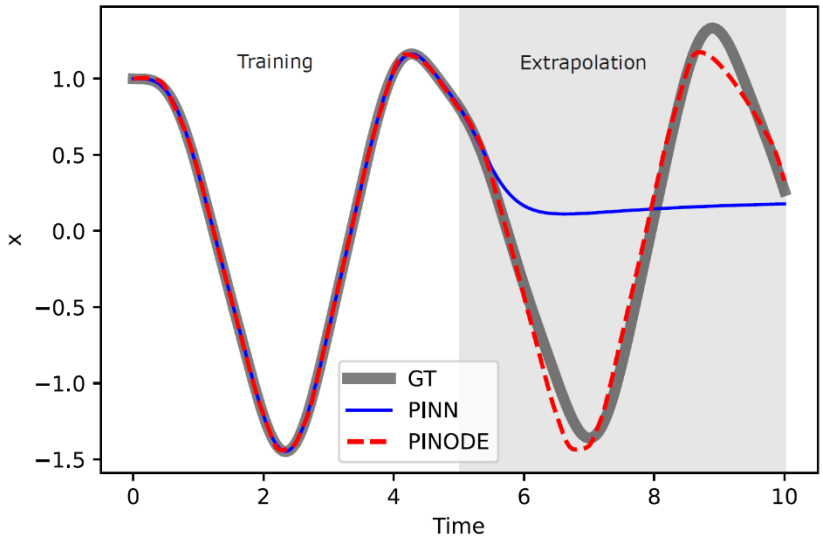


Figure 57. Comparison of the predicted positions of the mass on the x-axis from the training and extrapolation using the PINN and PINODE methods with the ground truth (GT).

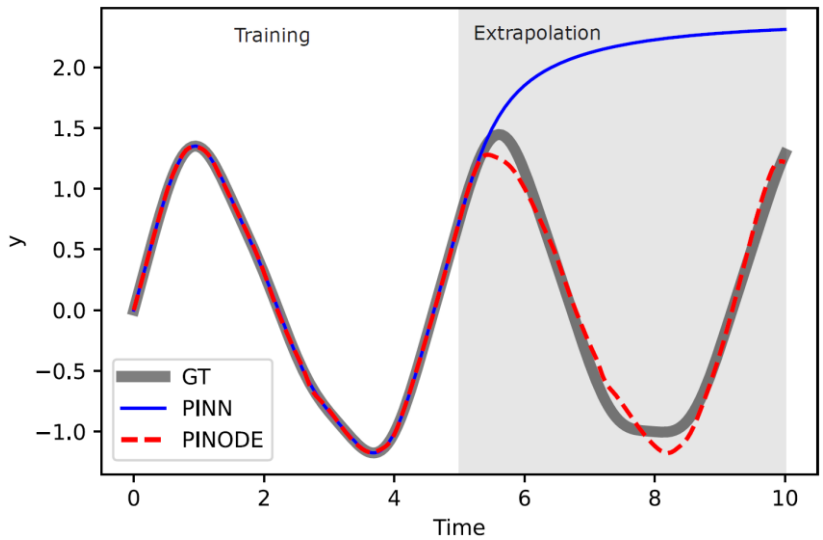


Figure 58. Comparison of the predicted positions of the mass on the y-axis from the training and extrapolation using the PINN and PINODE methods with the ground truth (GT).

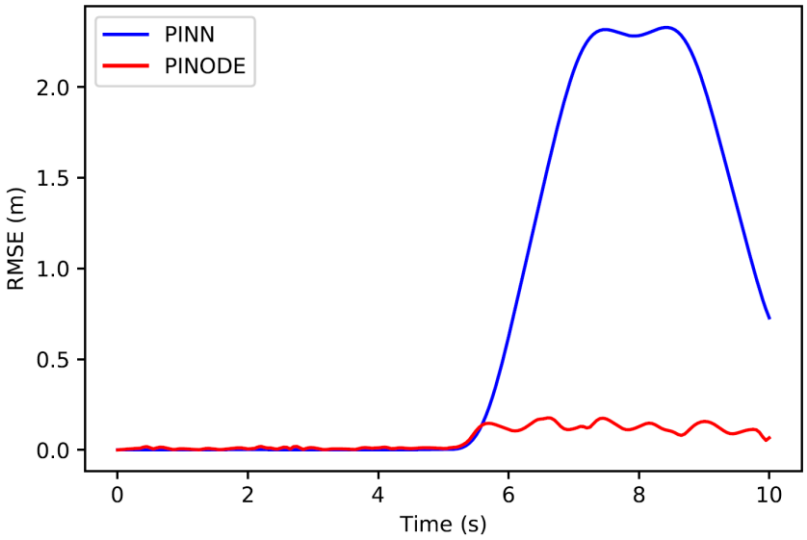


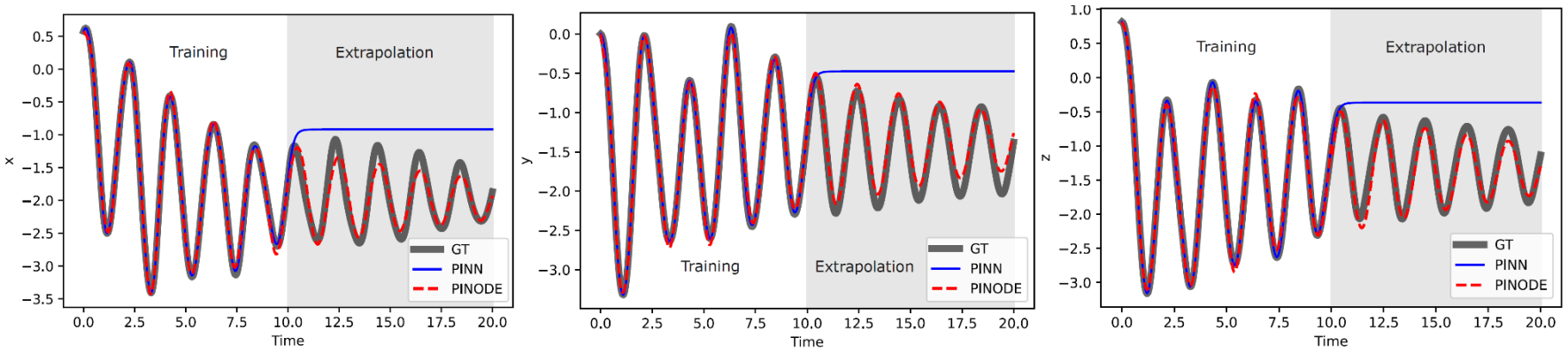
Figure 59. Plotting the root mean square errors (RMSE) over time steps when predicting positions of the mass using the PINN and PINODE methods.

5.3.2 3D mass-spring system

The predicted trajectories from the training set using PINN method achieved the RMSE of 0.012 m, L^2 error of 0.978% and the PCC of 0.999. These metrics suggest a high level of accuracy and correlation between the predictions and the actual values. On the other hand, the predictions from the training dataset using PINODE model returned slightly higher errors, with the RMSE of 0.069 m, L^2 error of 5.810% and the PCC of 0.997. When extrapolating the predictions from 10 s to 20 s, the PINODE yielded the RMSE of 0.168 m, L^2 norm relative error about 12%, the PCC of 0.951. Meanwhile, the PINN model showed a larger increase in errors, with the RMSE of 0.975 m, L^2 error of 57% and a lower PCC of 0.392. These metrics indicate that the PINN model has struggled more with accurate predictions during extrapolation compared to the PINODE model. Table 7 shows these performance metrics. Figure 60, Figure 61 and Figure 62 depict the plotting the predicted trajectories of three nodes according to the x-axis, y-axis and z-axis from the training and extrapolation using PINN and PINODE with the ground truth. In Figure 63, we can observe that the use of PINN method returned the RMSE significantly higher in the time [10, 20] s compared to the PINODE. The PINN has limited potential in extrapolation for prediction when faced with new initial conditions compared to PINODE.

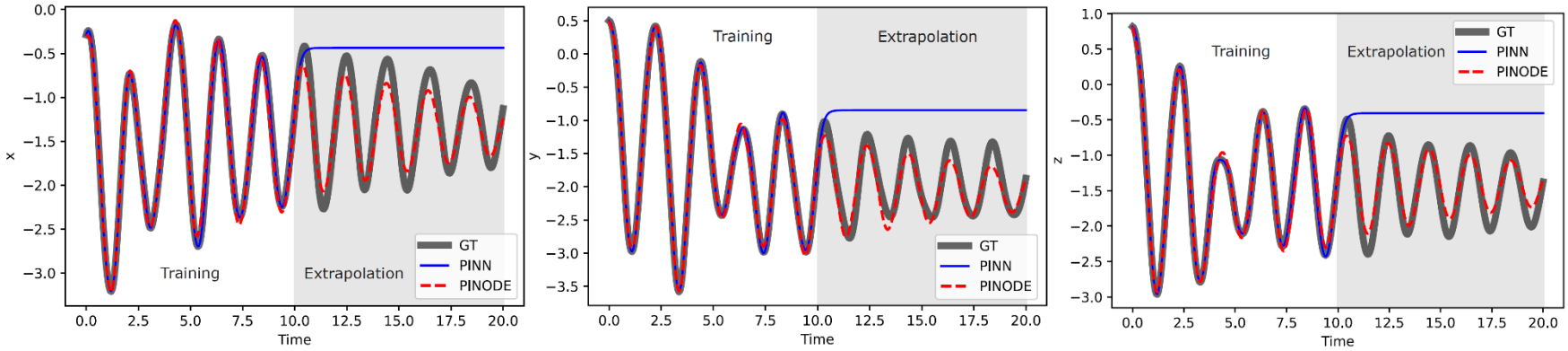
Table 7. Comparison of the performance metrics of the predicted motions of the 3D mass-spring system from the training during 10s and the extrapolation from 10s to 20s using the PINODE and PINN models.

	PINODE			PINN		
	RMSE (m)	L^2 (%)	PCC	RMSE (m)	L^2 (%)	PCC
Training	0.069	5.810	0.997	0.012	0.978	0.999
Extrapolation	0.168	12.099	0.951	0.975	57.102	0.392



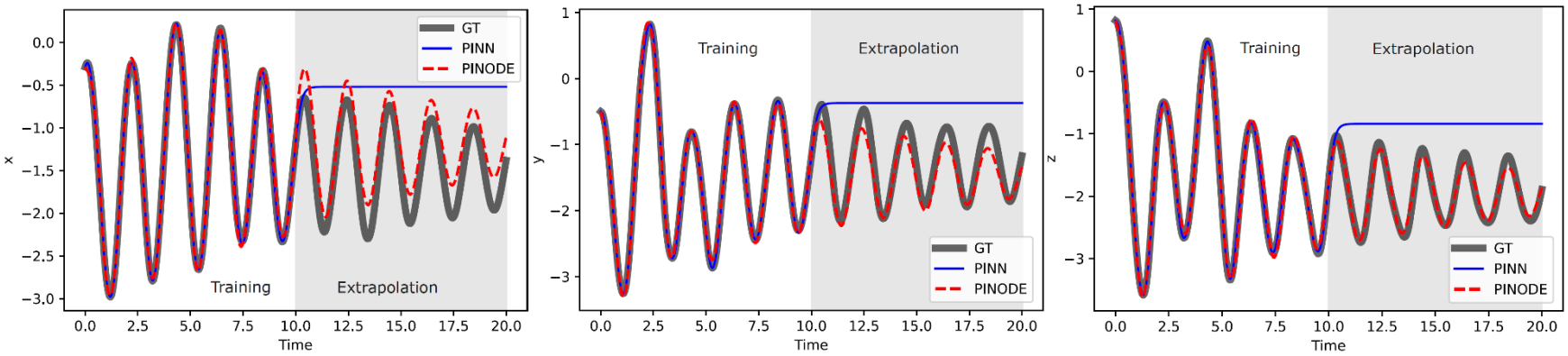
a) b) c)

Figure 60. Plotting the predicted positions of the node 1 on the a) x-axis, b) y-axis, c) z-axis from the training and extrapolation using the PINN and PINODE methods with the ground truth.



a) b) c)

Figure 61. Plotting the predicted positions of the node 2 on the a) x-axis, b) y-axis, c) z-axis from the training and extrapolation using the PINN and PINODE methods with the ground truth.



a) b) c)

Figure 62. Plotting the predicted positions of the node 3 on the a) x-axis, b) y-axis, c) z-axis from the training and extrapolation using the PINN and PINODE methods with the ground truth.

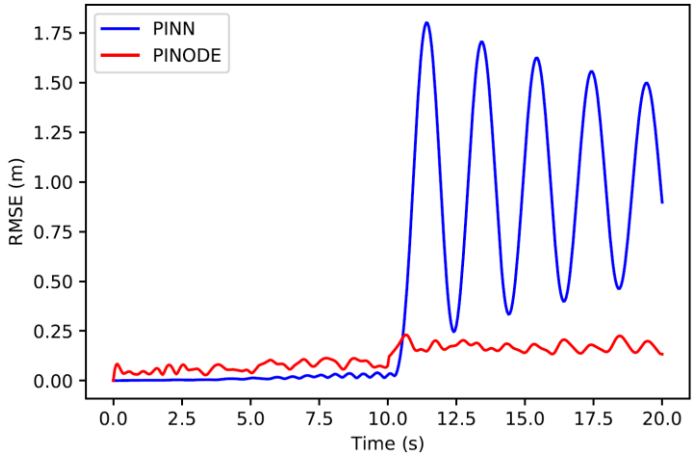


Figure 63. Comparing the root mean square errors (RMSE) over time steps when predicting positions of the three masses using the PINN and PINODE methods.

5.3.3 3D cantilever beam

The predicted displacement of the cantilever beam using the PINODE model performed remarkable accuracy in both training and extrapolation phases. During training, the RMSE was 2.625×10^{-5} m that indicated a high precision in capturing the beam deformation. The L^2 norm relative error was 0.077%. Even during extrapolation, the model maintained its precision with the RMSE of 1.057×10^{-5} m and L^2 error of 0.031%. Furthermore, the high PCC of 0.999 in both training and extrapolation underscored the strong linear relationship between the predicted and actual data. Table 8 depicts these metrics. Figure 64 shows that the predicted deformations of a node aligned well with the ground truth along the z-axis during 5-second training and extrapolation processes. The displacement of an edge beam is illustrated in Figure 65. The deformation is visualized in the color scale and 3D view in Figure 66 and Figure 67.

Table 8. Root mean square error (RMSE), L^2 norm relative error and Pearson correlation coefficient (PCC) from the predictions on the training set and the extrapolation using the PINODE model.

	PINODE		
	RMSE (m)	L^2 (%)	PCC
Training	2.625×10^{-5}	0.077	0.999
Extrapolation	1.057×10^{-5}	0.031	0.999

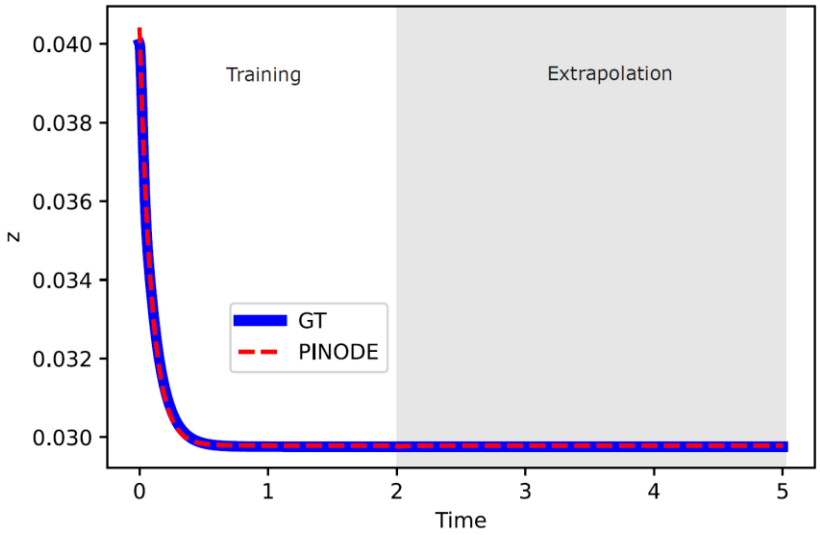


Figure 64. Plotting the displacement of a node in the beam over 5s following z-axis using the PINODE method.

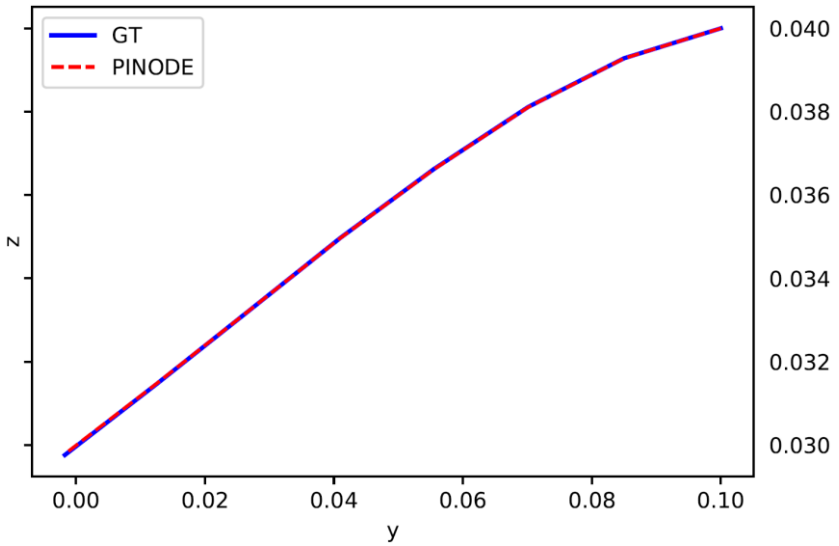


Figure 65. Comparing the displacement field predicted by the PINODE model with the ground truth (GT).

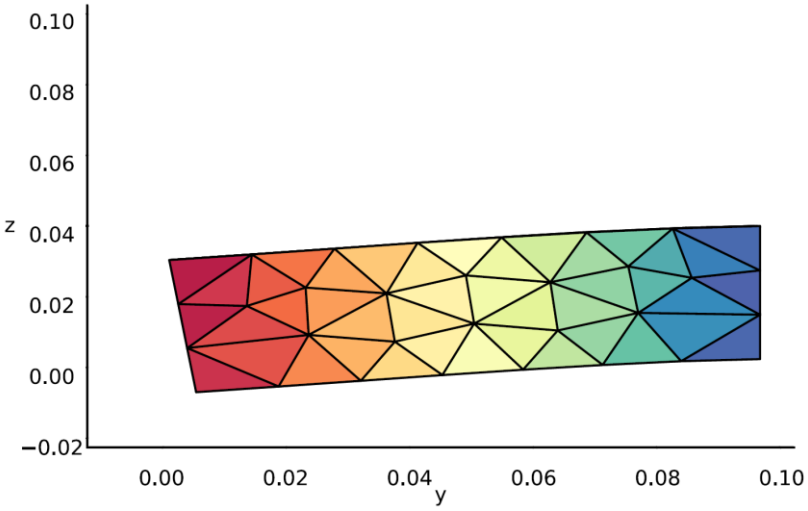


Figure 66. Deformation of beam under the gravity predicted by the PINODE method.

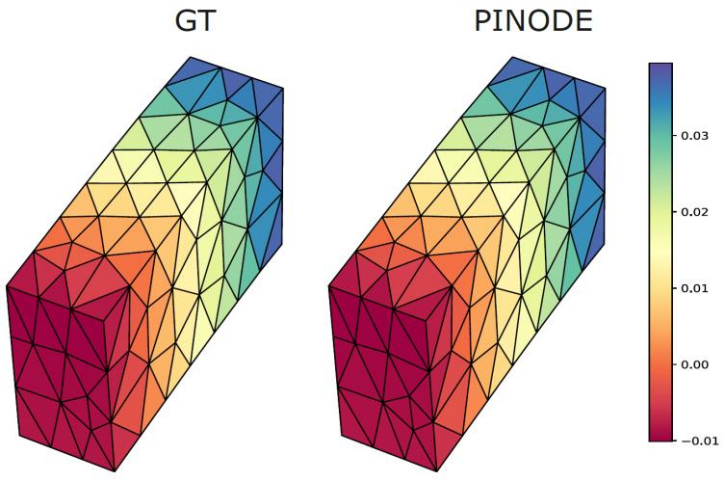


Figure 67. 3D view of the beam deformation under the gravity of the ground truth (GT) and predicted by the PINODE method.

5.3.4 3D uterus soft tissue model

The training phase of the PINODE model yielded promising results with the RMSE of 0.509 $\times 10^{-3}$ m and L^2 norm relative error of 0.349%. During the extrapolation phase, there were increases in the RMSE of 6.404×10^{-3} m and L^2 error of 6.357%. These metrics indicate an accurate representation of the proposed model, presented in Table 9. Figure 68 describes the predicted displacements on the training set at 0.5 s and 1 s. The extrapolation of the uterus displacements at 1.5 s and 2 s is shown in Figure 69 and Figure 70 displays a comparison between the visualization of deformation at the time 2 s using the PINODE model and corresponding ground truth. An interesting observation in this example is that the PINODE extrapolated displacements within just in 1 s, while conventional numerical method required 1 hour.

Table 9. Root mean square error (RMSE), L^2 norm relative error and Pearson correlation coefficient (PCC) from the predicted uterus soft tissue deformation on the training set and the extrapolation using the PINODE model.

	PINODE		
	RMSE (m)	L^2 (%)	PCC
Training	0.509×10^{-3}	0.349	0.999
Extrapolation	6.404×10^{-3}	6.357	0.996

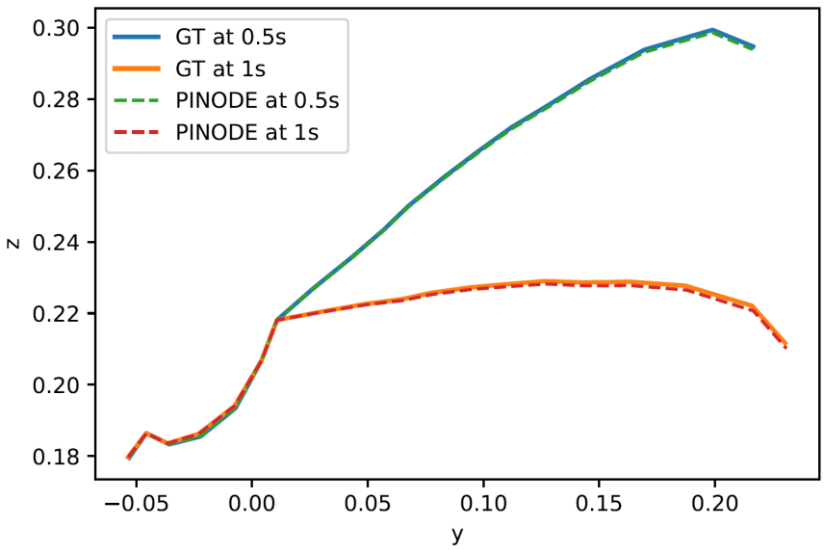


Figure 68. Comparison of the predicted displacements on the training set at 0.5 s and 1 s using the PINODE model with the ground truth.

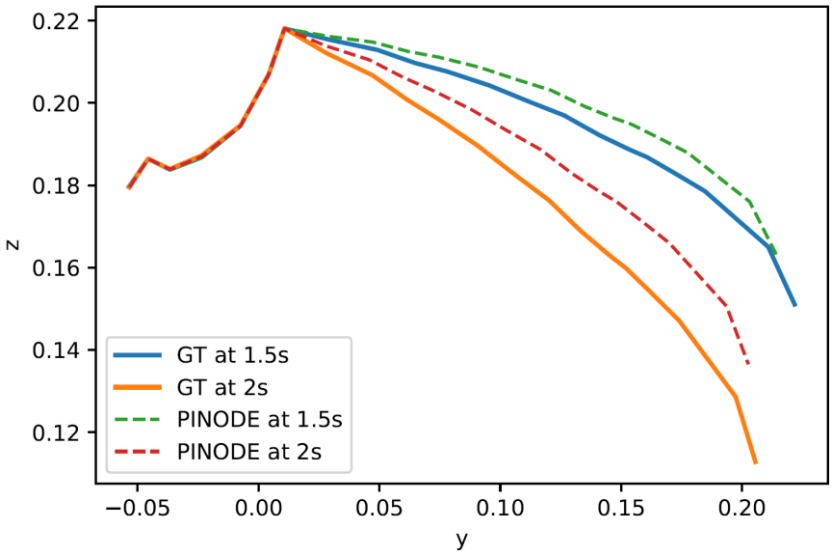


Figure 69. The PINODE model extrapolates the uterus displacements at 1.5s and 2s.

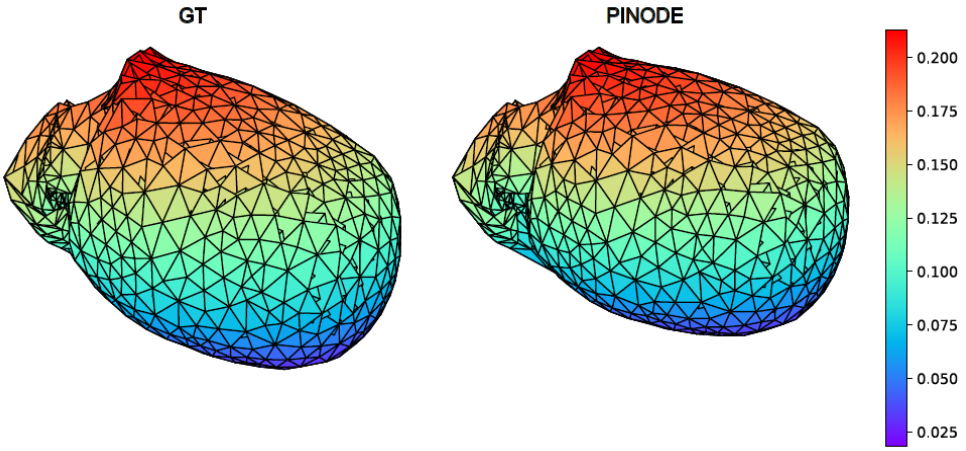


Figure 70. Visualization of the uterus soft tissue deformation at the time 2s using the PINODE model compared to the ground truth.

5.4 Discussion

Soft tissue deformation has been commonly described using traditional numerical methods like finite element method [154] or mass-spring model [159]. The use of a mesh configuration makes the solution convergence and numerical instabilities challenging for complex material definition and loading conditions. Besides, deep learning has been used to overcome these challenges for real-time soft tissue deformation [131]. Traditional deep learning approaches based essentially on a relevant and representative database to obtain an accurate prediction. Recently, physics-informed deep learning approach has been intensively investigated to incorporate physical knowledge to enhance the physical predicting capacities of the deep learning models. We presented in this study the use of vanilla physics-informed neural network (PINN) and physics-informed neural ODE (PINODE) for simulating different dynamic systems including 2D/3D mass-spring systems and the deformation of cantilever beams and soft tissue deformation. Obtained results showed encouraging accuracy levels from the training and extrapolation phases. In particular, the PINODE models showed strong potential in

extrapolating the motions of systems with new initial conditions compared to PINN models. Utilizing PINODE for extrapolation, the beam achieved astonishing root mean squared error (RMSE) of 1.057×10^{-5} m and L^2 relative error of 0.031% for the time from 2s to 5s. The extrapolating of soft tissue uterus deformation in [1, 2] s obtained an RMSE of 6.404×10^{-3} m and L^2 error about 6%. In general, the Pearson correlation coefficient (PCC) for most cases approached an impressive value of 0.99. Regarding the uterus soft tissue deformation, a noteworthy observation is that PINODE achieved extrapolated displacements in just 1 second, whereas the traditional numerical method took 1 hour for the same task.

In addition, Bazmara et al. [230] introduced a framework for PINN applied to the nonlinear bending analysis of 3D beams. The study leveraged the inherent physical principles governing a 3D functionally graded beam on a Winkler-Pasternak foundation and advantages in machine learning. The model included 5 hidden layers, 100 training points, a learning rate of 0.0001 and Adam optimizer. The training was processed through 60000 epochs. Notably, the PINN framework achieved predictions of bending up to 37 times faster than finite difference method. Wu et al. [231] enhanced the PINN through a residual-based adaptive distribution to solve 1D wave equation over a time span of 1s. The model was trained with 100000 iterations and incorporated 2000 residual points. The results showcased a remarkable achievement as the L^2 relative error was about 0.1%. Lai et al. [232] explored a novel approach to structural identification using PINODE incorporating domain knowledge in structural dynamics. The study investigated the free vibration scenario of 4 degrees of freedom dynamical system with masses and springs. The network was trained in 0 s to 6 s, followed by extrapolation up to 12 s. The number of iterations was 3000. The use of PINODE demonstrated superior performance in terms of faster convergence and higher precision compared to the direct implementation of NeuralODE. Our proposed PINN frameworks have demonstrated notable performance according to the literature when solving dynamic problems, requiring only the strong form of physical systems and small datasets. Particularly, our study has effectively addressed complex problems of 3D real-time soft tissue deformation with significant degrees of freedom. These cases included 357 degrees of freedom for the cantilever beam and 4041 degrees of freedom for the uterus systems. Another notable strength of our models lies in their rapid and accurate extrapolation capabilities, adeptly adapting to new initial conditions, thus offering promising avenues for future research and application in various domains.

The approach of PINN smoothly combines data with mathematical physics models, even in situations where understanding is partial and dimensions are high. Regression methods based on neural networks provide efficient and mesh-free implementations. The structure of PINN closely resembles that of conventional neural networks, with input, hidden, and output layers. Nonetheless, their unique characteristic is evident in the loss function, which incorporates terms aimed at enforcing the network's adherence to the governing physical laws and equations. PINNs are less susceptible to overfitting due to their incorporation of physical principles. They are applied in diverse issues, enhancing data mining processes and offering valuable insights into intricate scientific challenges. PINODE methods surpass the capabilities of the PINNs in scenarios where explicit formulations of ODEs are absent and there is limited measurement data. Although the cumulative root mean squared error and L^2 relative error for PINODE experiences a slight increase during extrapolation, it notably outperforms PINN models.

The design of NeuralODE is characterized by a continuous depth that exhibits an ability to maintain consistent memory costs throughout computational processes. Unlike traditional neural networks, NeuralODE adjusts its assessment approach for each input and can explicitly

exchange numerical accuracy for increased speed. The use of ODE solvers for optimizing backpropagation stands out due to its ability to hold on to a constant memory footprint, a stark departure from other techniques where memory consumption typically scales linearly with model complexity. By treating hidden layers as a continuous sequence rather than discrete ones, NeuralODE offers a significant advantage in terms of speed and memory efficiency.

Moreover, the use of the Julia programming language yielded advantages in terms of speed and computational cost. Julia was specifically outlined for scientific computing, offering a comprehensive array of packages that facilitate diverse applications in science, mathematics and machine learning. The primary attraction of Julia lies in its remarkable speed. Julia is known for its swiftness, employing just-in-time compilation, reducing the likelihood of code interpretation multiple times and minimizing overhead. Julia surpasses existing data programming languages like Python. A drawback of Python revolves around its speed. As a dynamically typed interpreted language, Python tends to be slower in comparison to compiled languages like C and Java. The flexibility of Python with respect to data types contributes to a higher consumption of memory.

Physics-informed neural networks exhibit significant promise for modeling dynamic systems governed by ordinary differential equations. Nevertheless, it is essential to acknowledge that they are accompanied by various limitations that warrant careful consideration. Vanilla PINN models employ deep networks composed of fully connected layers along with a specific variant of gradient descent. The hyperparameters are manually adjusted and the learning phases depend on the sample size and issues. Consequently, the training of these models may encounter issues like gradient vanishing, particularly posing challenges in handling 3-dimensional problems and resulting in potentially slow convergence. Keep in mind that PINNs encompass different terms in the loss function. The relative weights assigned to these terms significantly impact the predictions. Currently, there are no established guidelines for the optimal selection of these parameters. Another limitation of the vanilla PINN models relates to the extrapolation ability. To predict values from new initial conditions, PINNs need to train a new neural network. This requirement arises from the inherent limitation that the design of vanilla PINNs cannot grasp the fundamental works of a specific phenomenon, thereby restricting their ability to generalize.

Though physics-informed neural ODE models have shown promise in simulating intricate dynamical systems, these novel approaches encounter some difficulties. The training process of PINODE is challenging due to its continuous-depth structure. Therefore, PINODE models also encounter the limitations related to convergence and are dependent on both sample size and problem characteristics. Compared to PINN, PINODEs often have more complicated designs and require more work to implement. Despite their setup difficulty, PINODEs show promising potential for accelerating real-time computation of dynamic systems and demonstrate better generalizability compared to vanilla PINN.

Finally, the focus of future studies would be on investigating dependable learning methods to ensure secure predictions in extrapolation stage. Considering the application of deep operator networks (DeepONet) [233] may a viable option. The foundation of DeepONet lies in the universal approximation theorem for operators, allowing the precise and efficient learning of operators while minimizing generalization errors. Additionally, DeepONet has the capability to be trained using sparse labeled datasets and has the loss function established differential equations. Wang et al. [234] applied PINN based DeepONet in solving issues like Burger's equation, diffusion reaction. The outcomes showed notable gains in data efficiency, predictive

precision and generalization capability. From an applicative point of view, the current uterus soft tissue model is still limited in its representation of the various forces involved in the labor and delivery process, particularly those exerted by maternal contractions, pushing, and obstetric interventions, such as pulling to aid in the delivery. Achieving a comprehensive real-time interactive childbirth training tool necessitates a richer database encompassing real scenarios and a wide array of forces, external loads and soft tissue contact behaviors. The present study pays the way to this objective. Moreover, it should consider the incorporation of maternal hormonal and mechanical signaling, which leads to the remodeling of soft tissues in the vagina and perineum during pregnancy, resulting in increased vascularity and reduced stiffness of connective tissue. Integrating these factors will enhance the fidelity and effectiveness of childbirth simulation models.

Chapter 6

General Discussion

6.1 Thesis overview

Minimizing maternal and fetal mortality cases is crucial for sustainable healthcare systems [1], [2], [9], [10], [11]. Central to this endeavor is the prevention of stillbirths and ensuring that mothers receive steadfast assistance from qualified healthcare professionals throughout their pregnancy and delivery. This commitment not only enhances healthcare outcomes but also fosters a supportive environment where maternal and fetal health is prioritized and safeguarded. The development of a skilled medical workforce with the knowledge and skills to reduce the rate of stillbirths and maternal fatalities is essential to achieving this aim. Over the years, numerous methods have been developed to address these challenges.

Previously, one method utilized for addressing childbirth challenges involved analyzing observational data collected over time. However, relying solely on radiological pelvimetry to predict delivery methods has proven limited [14], [15]. Studies have underscored its constraints, particularly in anticipating whether an infant can navigate the pelvis without encountering cephalopelvic disproportion. While radiological pelvimetry indicates pelvic health, it does not guarantee successful traversal for the fetus. Another method was the use of physical birth simulators for healthcare professionals to gain practical experience in delivery training [16], [17]. Nevertheless, their ability to offer a thorough understanding of birth-related injuries was limited. These simulators lacked the ability to depict patient-specific data, as they were designed for average-sized women and fetuses. Furthermore, they made no effort to verify the mechanical properties of the pelvic soft tissue. Childbirth computational models based on numerical methods have been investigated to understand the impact of the process on the mother and fetus [18], [19], [20]. These models have provided insights into fetal and maternal damage during labor, addressing the shortcomings of physical birthing models. Advancements in computer technology and numerical modeling have enabled the development of more realistic and advanced childbirth models. However, there are several restrictions on the present research on birth simulations:

1) Current studies on fetal models have generally neglected to incorporate full-body representations with joint articulations, focusing instead on the head and neck.

2) The utilization of mesh-based methods to solve the motion of soft tissue remains challenging in time consumption for computation and processing.

3) There is still a deficiency in using deep learning methods to predict real-time pelvic soft tissue deformation.

Therefore, this thesis aims to develop models for fetal representation and real-time soft tissue deformation. These models will contribute to creating an advanced decision support tool for childbirth training and simulating complications.

6.2 Main contributions

The thesis presented three main contributions:

- The first contribution was the automated and fast segmentation of newborn skeletons into distinct parts employing a model based on generative adversarial networks and 3D point cloud data.

- The second contribution was the accurate and rapid prediction of soft tissue deformations applying recurrent neural networks (i.e. long short-term memory neural networks) integrated with a learning strategy based on principal component analysis.

- The final contribution was developing a cutting-edge model to simulate real-time soft tissue deformations, based on the physics-informed neural networks and neural ordinary differential equations.

The deep learning models proposed in the thesis have undergone rigorous numerical validation demonstrating a commendable level of precision when compared to conventional methods. Through the utilization of advanced techniques, the thesis successfully enhanced accuracy, minimized computational costs, improved simulation speed, and toward a next-generation tool for childbirth training simulation. These advancements collectively contribute to a more comprehensive understanding of childbirth dynamics. The results not only validate the efficacy of the proposed deep neural models but also highlight their potential to revolutionize the field by providing a more efficient and precise approach to studying and simulating childbirth processes.

6.2.1 Skeleton part segmentation of newborns from 3D point clouds

Childbirth simulations have been studied in order to predict and prevent difficult delivery issues. The reconstruction of the maternal pelvic model, which consists of a comprehensive fetal model with articulated joints, is important for therapeutic purposes. However, it is difficult and time-consuming to segment the various bones using classical image processing approaches. The aim of the study in Chapter 3 is to develop and evaluate a generative adversarial network to automatically and fast segment the bone elements of the complete neonatal skeleton.

A database of 124 newborn CT images was collected and segmented. Each 3D reconstructed skeleton was divided into 23 distinct bony segments. We proposed the generative adversarial network based on PointNet to perform the automated segmentation directly on the 3D point clouds. Our method was compared to the pointwise convolutional neural network to demonstrate its accuracy and efficiency. The GAN model produced highly accurate results with an IoU of 93.68% ± 7.37%, a Dice of 96.56% ± 4.41%, and an accuracy score of 96.72% ± 3.56%, compared to 72.30% ± 5.10% for IoU, 83.82% ± 3.44% for Dice and 84.81% ± 3.25% for accuracy respectively by the pointwise convolutional neural network. With the best GAN algorithm, we obtained the best prediction with 99.63%, 99.81%, and 99.98% for the means of IoU, Dice and accuracy score, respectively. Meanwhile, the worst case returned the mean IoU

of 65.81%, the mean Dice of 79.37%, and the mean accuracy score of 80.03%. In addition, our model behaved better on skeletons in anatomical postures than ones in fetal positions.

The study opens new avenues for fast and accurate 3D part segmentation of the newborn 3D skeleton. In the future, further study should focus on segmenting fused bones like vertebrae and integrating the whole articulated skeleton into the maternal pelvic model to simulate complex vaginal delivery and perform associated preventive actions.

6.2.2 A novel deep learning-driven for predicting pelvic soft tissue deformations

Soft-tissue dynamics plays an essential role in the mechanical functions of the human body. Numerical approaches using finite element modeling and mass-spring framework have been currently used to estimate the biological soft tissue dynamics. However, these approaches still have important computational costs due to the use of a mesh configuration in the formulation of the dynamic equilibrium equation and unstable convergence issue.

We present in Chapter 4 a novel approach based on the deep learning framework to predict the deformation of soft tissues. In particular, the Long Short-term Memory (LSTM) neural network and deep neural network (DNN) were used to deal with high-frequency oscillation signals. Different learning strategies (with and without data dimension reduction) were also applied. A simulation-based database was generated using our HyperMSM model for training and testing purposes. The application of the proposed approach on the childbirth simulation was addressed.

Using the root mean square error, the LSTM- and DNN-derived deformation deviation range from 0.139 mm to 1.062 mm (0.266% to 2.028%) for both training and testing processes. The Pearson correlation coefficient of 0.994 – 0.999 demonstrates the strong similarity between predicted outputs and ground truth data. The best predicted case from the test database produced mean errors of 0.103 mm ± 0.116 mm using LSTM, and 0.107 mm ± 0.083 mm using DNN. For the worst prediction, using LSTM resulted in a mean error of 0.845 mm ± 1.214 mm, while the DNN-based deformation forecast has a mean error of 0.775 mm ± 1.041 mm.

This work showed the capacity of the deep learning neural networks to predict complex physiological signals of the human body functions. As a perspective, this approach will be coupled with the Hololens device toward a novel interactive childbirth training tool with real-time feedback on the soft tissue deformations.

6.2.3 Novel coupled state-of-the-art methods to simulate the real-time deformation of soft tissue

Traditional numerical methods of the soft tissue deformations, based commonly on a mesh configuration, incur large processing costs and difficulties associated with unstable convergence. Physics-informed neural networks (PINN) have recently been recognized as an effective approach for solving differential equation problems. In Chapter 5, we developed and evaluated a suitable predictive model of the soft tissue deformation based on PINN and neural ordinary differential equations (NeuralODE) coupled with the mass-spring system (MSS) modeling framework.

Theoretical basis of the proposed approach was established using the PINN, NeuralODE and MSS formulations. Then, we evaluated our novel approach on the 2D and 3D mass-spring

systems as well as a 3D cantilever beam. Finally, a 3D uterus model deformation under gravity loading was simulated and evaluated.

The best extrapolation phase was in the experiment of the cantilever beam that achieved a stunning root mean squared error (RMSE) of 1.057×10^{-5} m and L^2 relative error of 0.031% for the time from 2s to 5s. The extrapolating aspect of soft tissue uterus deformation from 1s to 2s returned an RMSE of 6.404×10^{-3} m and L^2 error about 6%. Overall, the Pearson correlation coefficient (PCC) for most cases was 0.99. The physics-informed neural ordinary differential equations models showed strong potential in extrapolation compared to vanilla PINN.

We designed and assessed an outstanding model to simulate deformations of soft tissue using the physics-informed neural networks and neural ordinary differential equations. This achievement opens new avenues in the real-time soft tissue deformation during the childbirth simulation toward a next-generation decision support tool for childbirth training and complication simulation.

6.3 Limitations

While our methods have yielded promising results, there are some things to be aware of. We have examined every aspect of our methodology, identifying several limitations discussed in each part.

We proposed the GAN-based PointNet model for segmenting skeletons of newborns from 3D point clouds. The drawback of PointNet was its inability to collect local meaning between points. PointNet has separately learned features of each point. Consequently, numerous methods have to depend on expensive neighbor searching strategies, limiting the success of the current method. While GAN-based models have opened up new possibilities for 3D point cloud segmentation, there was a certain difficulty with the cost and sophisticated training setup. Another limitation was that the prediction relied on the fetal postures during scanning. A greater degree of precision was attained when the individuals were positioned physiologically. There was poor segmentation when infants' arms were positioned on their chests. Additionally, it was challenging to differentiate between fused organs like vertebrae. Moreover, the limited amount of data was a challenge because of the difficulties in gathering fetal data. Data collection is complicated since the unborn fetuses are usually placed within the mother's uterus.

The pelvic soft tissue models developed in this thesis have been constrained by their inability to fully capture the diverse forces at play during the labor and delivery process. Our experiments were predominantly reliant on gravity or external forces, thus limiting the scope of our findings. We used straightforward node-to-triangle contact models to control the interface between the fetal body and the mother's uterus. The absence of complicated contact models was a drawback in our investigation. Another limitation regarding the scope of delivery scenarios explored. We only focused on the normal birthing process, where delivery occurs naturally through the vagina without external intervention.

An additional constraint concerned the application of recurrent neural networks to predict soft tissue deformation in real-time. In our study, the accuracy level provided by long short-term memory networks was comparable to that of traditional deep neural networks. A potential explanation may lie within our problem formulation. Predicting the next time step solely based on the preceding seems not to align well with the philosophy of recurrent neural networks.

Although there was great potential for simulating dynamic systems driven by ordinary differential equations with physics-informed neural networks, there were several drawbacks to consider. Usually, these models use fully connected deep neural networks that were trained using a particular type of gradient descent. Fine-tuning hyperparameters may be difficult, and sample size and computing difficulties have a big impact on how successful learning phases are. Thus, gradient vanishing was one of the difficulties that PINN models may face during training, especially when dealing with 3D issues. In addition, the loss function of PINN included some terms and the relative weights given to these items impacted predictions. The lack of defined recommendations for optimizing these factors, despite their potential, highlighted the necessity for more studies. The extrapolation capacity of the conventional PINN models was another drawback. PINNs require building a new neural network in order to predict values from novel beginning conditions. This stipulation stems from the intrinsic constraint that the architecture of vanilla PINNs is unable to comprehend the essential components of a particular phenomena, which limits their generalizations. In physics-informed neural ordinary differential equations models, the limitation concerned the convergence. This problem was due to the continuous-depth structure, which made it difficult to interpret the data's subtleties and provide trustworthy predictions. Thus, the convergence of PINODE models mostly relied on the training size and the characteristics of the issue.

Chapter 7

Conclusions and Perspectives

7.1 Conclusions

This thesis represented a groundbreaking advancement in the realm of realistic fetal representation and real-time soft tissue deformation modeling. By integrating cutting-edge techniques and innovative methodologies, it opens new avenues in the field of healthcare, promising to revolutionize childbirth training and complication simulation. With a focus on accuracy and precision, this research laid the foundation for the development of a next-generation decision support tool. These tools not only enhance the effectiveness of childbirth training but also facilitate the simulation of various complications, providing invaluable insights for medical professionals. By bridging the gap between theory and practice, this thesis marked a significant step forward in improving maternal and neonatal healthcare outcomes.

Through this thesis, we proposed a deep learning approach for segmenting the newborn skeleton into 23 distinct bony segments directly from the 3D point clouds. The results demonstrated that our generative adversarial network based on the PointNet model was fast and highly precise for the segmentation of the different bony parts. The proposed GAN model used the advantage of T-net in PointNet to build the generator and 1D convolutional neural network to build the discriminator. The segmentation outcomes were achieved in just 5 seconds compared to 30 to 50 minutes using software like 3DSlicer. Thus, the research has the potential to pave the way for the creation of reliable childbirth simulations with detailed articulated skeletons.

Following up, a novel deep learning-driven approach for predicting the pelvis soft tissue deformation using recurrent neural and classical deep neural networks was developed and evaluated. Obtained outcomes showed a strong correlation between predicted outcomes and ground truth values. The combination method of long short-term memory neural networks, deep neural networks, and principal component analysis-based learning also was proposed. The use of the principal component analysis method reduced significantly the dimension of data that assisted networks save computational costs, but kept the accuracy. The proposed approach can be a promising solution, which is fast, stable, accurate and without mesh configuration, to predict and track the displacement of pelvic soft tissues.

Furthermore, we developed physics-informed neural networks (PINN) and physics-informed neural ordinary differential equations (PINODE) to predict deformations of biological soft-tissue toward a real-time interactive simulation solution. We evaluated these methods on numerical examples of 2D and 3D mass-spring systems, 3D cantilever beams, and 3D soft tissue uterus. The predicted results from the training dataset returned very good accuracy in terms of root mean square error, L^2 norm relative error and Pearson correlation coefficient when using both two methods. PINODEs demonstrated superior performance in extrapolations when compared to PINNs, particularly excelling in extrapolating the behavior of a cantilever beam

and the soft tissue of the uterus. Overall, the proposed models demonstrated exceptionally fast prediction speeds, typically within a second, in comparison to traditional numerical methods.

Although our proposed methods have shown promising results, it is important to acknowledge certain limitations. Segmenting newborn skeleton parts proved challenging due to fused bones, while a limited dataset further complicated the process. Existing pelvic soft tissue models lacked a comprehensive representation of the diverse forces inherent in labor and delivery. The childbirth simulation models were simplistic, focusing solely on normal circumstances and failing to incorporate the complexities found in real-world cases. Furthermore, to govern the interaction between the fetal body and the maternal uterus, we utilized simplistic node-to-triangle contact models. Therefore, future works will aim to address these drawbacks to develop a more comprehensive and accurate model that can effectively simulate the dynamics of childbirth and offer insights into the physiological processes involved.

7.2 Perspectives

Efforts in the future will focus on refining outcomes and overcoming present constraints.

The accuracy of skeleton part segmentation in newborns hinges on the depiction of fetal postures. One notable challenge lies in effectively distinguishing between fused organs. Future studies should focus on segmenting fused bones like vertebrae and merging the whole articulated skeleton into the maternal pelvic model to perform complex delivery simulations and associated preventive actions. Additionally, research efforts should be directed toward collecting more data to refine the existing and enhance segmentation accuracy. Moreover, there should be a focus on investigating solutions that involve meticulous attention to spine articulation, particularly concerning the cervical vertebrae.

From a methodological point of view, we aim to investigate points that divide geometry into parts, and use them as a learning feature to improve the accuracy of the model. Moreover, edges and concave points could be combined to precisely delineate the borders between vertebrae to increase the segmentation accuracy for overlapping vertebrae. Furthermore, further study could be also investigated by using different variants of the GAN models such as deep energy GAN model for point cloud processing.

The current soft tissue model of the uterus has limitations in accurately capturing the complexities of childbirth dynamics. This thesis primarily concentrates on normal fetal delivery scenarios, showcasing its narrow scope in representing the diverse forces involved in childbirth. To address these shortcomings, future research endeavors plan to incorporate additional factors such as maternal contractions, pushing, and obstetric interventions like pulling to facilitate birthing. Additional research will involve integrating more intricate delivery scenarios, such as forceps-assisted deliveries or abnormal fetal positions during birth. This effort is aimed at improving the effectiveness of the tool in equipping healthcare professionals for a wider array of childbirth circumstances. Enriching the database to include a diverse range of real-world scenarios will ensure that the training tool faithfully mirrors the complexities encountered in clinical practice. Furthermore, the current models utilize a simplistic node-to-triangle contact model to govern the interaction between the fetal body and the soft tissue of the pelvis. To achieve a more authentic interaction, it is imperative to implement a more sophisticated contact model in future developments. This enhancement will enable a stronger representation of the intricate dynamics between the fetal body and pelvic soft tissue.

Additionally, future studies should explore the integration of maternal hormonal and mechanical signals into simulation models of birthing. Throughout pregnancy, these variables play pivotal roles crucial in the reshaping of the pelvic soft tissues such as the uterus, and vagina. Specifically, they contribute to increased vascularity and decreased stiffness of connective tissue, shaping the physiological landscape of childbirth. By incorporating these intricate mechanisms into simulation frameworks, a more comprehensive understanding of maternal and fetal interactions during labor and delivery can be achieved. This will improve the realism and efficacy of childbirth training resources.

Achieving high-fidelity pelvic soft tissue deformations in real-time necessitates accurate predictions during the extrapolation phase. Therefore, the focal point of forthcoming research will center on exploring dependable learning methodologies to guarantee secure prediction at this stage. Our strategy will study the utilization of neural operators-based physics-informed neural networks [223], advancing the capability for real-time childbirth simulation.

In childbirth simulation, essential components include a maternal pelvic model and a detailed fetal model with articulated joints. Future research aims to integrate these crucial models into the Hololens device [23], envisioning an enhancement of childbirth training skills through the creation of physiologically plausible delivery scenarios. By merging these models within the immersive environment of the Hololens, trainees will gain valuable hands-on experience and insights into the complexities of childbirth, fostering a more comprehensive understanding of maternal and fetal dynamics.

Publications

Journal articles:

- **Duyen H. Nguyen-Le**, A. Ballit, T-T Dao. *Physics-Informed Neural Ordinary Differential Equations and Mass-Spring System Modeling Framework for Predicting the Real-Time Soft-Tissue Deformations*. Engineering Applications of Artificial Intelligence (Submitted).
- **Duyen H. Nguyen-Le**, M. Ferrandine, D.P. Nguyen, V.D. Tran, H.D. Vo, T.N. Nguyen, T-T Dao. *Generative Adversarial Network for Newborn 3D Skeleton Part Segmentation*. Applied Intelligence 54, 4319–4333 (2024) http://dx.doi.org/10.1007/s10489-024-05406-0
- M. Ferrandini, D.P. Nguyen, **Duyen H. Nguyen-Le,** V.D. Tran, H.D. Vo, T.N. Nguyen, Tien-Tuan Dao. *Automatic part segmentation for full newborn skeleton-articulated geometries using geometric deep learning and 3D point cloud*. Int. J. of Biomedical Engineering and Technology (2024). In Press.
- **Duyen H. Nguyen-Le**, A. Ballit, T-T Dao. *A novel deep learning-driven approach for predicting the pelvis soft-tissue deformations toward a real-time interactive childbirth simulation*. Engineering Applications of Artificial Intelligence, 126, 107150 (2023). http://dx.doi.org/10.1016/j.engappai.2023.107150

Conference:

- A Ballit, **DH Nguyen-Le**, TN Nguyen, TT Dao (2022). Toward a deep learning-driven prediction of dynamic soft-tissue deformations for a real-time mixed-reality simulator of the childbirth processesVirtual Physiological Human Conf (VPH2022), 6-9 September 2022, Porto, Portugal (Oral).

Brief summary in French

Titre : Modélisation avancée du système d'accouchement à l'aide de différentes méthodes d'apprentissage profond : de la segmentation du squelette fœtal à la déformation des tissus mous en temps réel.

1. Introduction

L'Organisation mondiale de la santé (OMS) et ses partenaires [1] ont estimé que le taux mondial de mortalité maternelle en 2020 était de 223 pour 100000 naissances vivantes. Environ 287000 femmes sont décédées pendant et après la grossesse et l'accouchement. Cela correspond à près de 800 décès par jour, soit environ un toutes les deux minutes. Ce chiffre est plus d'un tiers inférieur à celui de l'année 2000, où l'on estimait à 446000 le nombre de décès maternels. Environ 2 millions de mortinaissances (c'est-à-dire des bébés nés sans signes de vie à partir de 28 semaines de grossesse ou plus tard) ont été signalées dans le monde en 2019 [2]. Des soins adéquats auraient pu potentiellement prévenir nombre de ces événements. Le taux mondial de mortinaissance s'élevait alors à 13,9 mortinaissances pour 1 000 naissances. Cela signifie qu'une naissance sur 72 se soldait par un décès fœtal, soit une occurrence toutes les 16 secondes. Au cours des deux dernières décennies, on ne dénombre pas moins de 48 millions de mortinaissances dans le monde. Cependant, il est important de noter que ce chiffre pourrait être sous-estimé car les mortinaissances sont souvent sous-déclarées. Un facteur contribuant de manière significative à ce taux de mortalité alarmant est le travail obstétrical compliqué. Dans de tels cas, le bébé ne parvient pas à sortir du bassin pendant le processus d'accouchement, malgré la contraction normale de l'utérus. Cette complication constitue une menace sérieuse pour la santé maternelle et infantile, soulignant le besoin de soins obstétricaux améliorés et d'interventions pour faire face et atténuer les risques associés au travail obstétrical compliqué.

Une méthode utilisée auparavant était l'analyse statistique des données d'observation collectées dans le passé. Les coefficients de corrélation entre une situation de travail difficile et des caractéristiques maternelles/fœtales particulières peuvent être obtenus en examinant des données décrivant une grande variété d'accouchements. De nombreuses études ont souligné les limitations de se fier uniquement à la pelvimétrie radiologique pour prédire la méthode d'accouchement [14], [15]. Cette technique ne permet pas d'anticiper si un nourrisson peut traverser avec succès le bassin mesuré sans rencontrer de disproportion céphalopelvienne. Elle indique simplement si le bassin est sain. La conformité de la tête fœtale a un effet déroutant dans la prédiction de la disproportion céphalopelvienne à travers une évaluation radiologique simple de l'interaction entre la tête fœtale et le bassin de la mère.

En revanche, l'utilisation de simulateurs de naissance physiques présente une méthode pour comprendre le processus d'accouchement. Ces simulateurs permettent aux professionnels de la santé d'acquérir une expérience pratique dans la formation à l'accouchement, ce qui est un aspect important de la préservation de la vie des mères et de leurs nourrissons. MamaNatalie [16] et Noelle [17] sont deux simulateurs de naissance courants, offrant un système d'enseignement complet pour former les étudiants en médecine. Cependant, les modèles physiques de naissance présentent plusieurs inconvénients en ce qui concerne la compréhension plus approfondie des mécanismes des blessures liées à l'accouchement. Tout d'abord, la capacité de ces simulateurs à représenter des données spécifiques au patient est limitée. Ils sont

généralement conçus pour imiter un scénario d'accouchement typique, qui comprend une femme de taille moyenne et un fœtus de taille moyenne. Il est difficile de modifier un simulateur pour qu'il corresponde aux mesures uniques d'un patient une fois qu'il est construit. Deuxièmement, les modèles physiques d'accouchement ont tendance à simplifier à l'excès la géométrie des tissus mous, tels que les muscles du plancher pelvien, sans chercher à valider les caractéristiques mécaniques des tissus mous. La simulation précise de la physiologie complexe sur des simulateurs physiques devient problématique.

Par conséquent, des modèles computationnels d'accouchement ont été explorés en utilisant des méthodes numériques pour étudier comment l'accouchement affecte les tissus des mères et des nouveau-nés [18], [19], [20]. Les approches ont le potentiel de résoudre certaines contraintes des modèles physiques d'accouchement, offrant une compréhension plus profonde des mécanismes sous-jacents aux blessures maternelles et fœtales pendant le travail et l'accouchement. Le développement de la technologie informatique et des logiciels de modélisation numérique ont conduit à la création de modèles computationnels de naissance plus réalistes et avancés, capables d'examiner des sujets complexes. Cependant, les études actuelles sur les simulations d'accouchement rencontrent certaines limites. La recherche s'est principalement concentrée sur les blessures du plancher pelvien subies par les mères pendant l'accouchement, le système pelvien maternel souvent représenté étant les muscles du levator ani. Les modèles fœtaux, dans de nombreux cas, se sont limités à inclure uniquement la tête, les articulations étant rarement prises en compte. Le processus d'accouchement a souvent été influencé par des forces prédéterminées. De plus, des matériaux présentant une rigidité extrêmement élevée ont été utilisés pour représenter les modèles fœtaux.

Ces dernières années, l'Intelligence Artificielle (IA) a connu des avancées remarquables, devenant un outil convaincant pour diverses procédures de soins cliniques. Ses potentielles applications dans le domaine médical ont suscité une attention croissante des milieux de la recherche, avec l'émergence de l'apprentissage automatique et de l'apprentissage profond comme des technologies indispensables. De diagnostics à la thérapeutique, en passant par la prédiction du pronostic et la gestion de la santé des patients, l'IA révolutionne la prestation des soins de santé. Dao [84] a utilisé un réseau de neurones à mémoire à court et long terme pour prédire les forces musculaires squelettiques à partir de données cinématiques articulaires pendant une période de marche. Ballit et Dao [85] ont également appliqué cette approche pour prédire le stress musculaire squelettique lors d'un comportement de contraction dynamique. Dans les simulations d'accouchement, la construction d'un modèle complet du système pelvien de la femme enceinte et du corps fœtal est importante pour faire avancer la compréhension des mécanismes de l'accouchement. Cependant, des défis scientifiques subsistent dans la représentation réaliste du fœtus, le coût computationnel et la vitesse de traitement adaptés pour déployer les simulations d'accouchement dans les pratiques cliniques courantes. Les objectifs de cette thèse de doctorat sont de relever ces défis.

La thèse est divisée en 7 chapitres. Le premier chapitre introduit le contexte des soins de santé, une vue d'ensemble des simulations d'accouchement, les applications de l'apprentissage automatique et de l'apprentissage profond dans les soins de santé, ainsi que les objectifs de la thèse. Le chapitre 2 passe en revue et explore l'état de l'art de la segmentation des parties du squelette 3D du nouveau-né et la prédiction de la déformation des tissus mous en temps réel. Nous proposons et évaluons un modèle basé sur un réseau antagoniste génératif (GAN) pour segmenter le squelette fœtal en parties distinctes, utilisant les données du nuage de points 3D du chapitre 3. Dans le chapitre 4, nous construisons et évaluons des modèles pilotés par

l'apprentissage profond pour prédire la déformation des tissus mous pelviens dans des simulateurs interactifs d'accouchement, impliquant l'intégration de réseaux neuronaux à mémoire à court et long terme (LSTM), de réseaux neuronaux profonds et de l'apprentissage basé sur l'analyse en composantes principales (PCA). Nous développons et évaluons le modèle de pointe pour simuler les déformations en temps réel des tissus mous en utilisant des réseaux neuronaux informés par la physique (PINN) et des équations différentielles ordinaires neuronales (NeuralODE) dans le chapitre 5. Le chapitre 6 discute des contributions clés et résume le travail réalisé. Le dernier chapitre présente les conclusions et les perspectives.

2. Réseau antagoniste génératif pour la segmentation des parties du squelette 3D du nouveau-né

La recherche sur les simulations d'accouchement vise à anticiper et à atténuer les complications lors de l'accouchement. Au cœur de ces travaux de recherche se trouve la reconstruction du modèle pelvien maternel, qui intègre un modèle fœtal détaillé avec articulations. Cependant, les méthodes traditionnelles de segmentation des structures osseuses diverses à l'aide de techniques classiques de traitement d'images sont ardues et chronophages. Ainsi, le chapitre 3 s'efforce d'introduire et d'évaluer un réseau antagoniste génératif (GAN) [71] adapté pour segmenter rapidement et automatiquement les composants osseux de l'ensemble du squelette néonatal, offrant une solution prometteuse à ce défi.

Le réseau antagoniste génératif est un modèle génératif de pointe capable de comprendre le schéma des données d'entraînement et de générer de nouveaux échantillons basés sur celui-ci. La découverte fondamentale du GAN est la procédure d'entraînement en tant que compétition entre deux réseaux neuronaux: un générateur pour créer de nouveaux échantillons plausibles et un discriminateur pour différencier quels échantillons sont faux et lesquels sont réels. Ces réseaux sont entraînés simultanément. La Figure 1 représente le schéma d'un GAN.

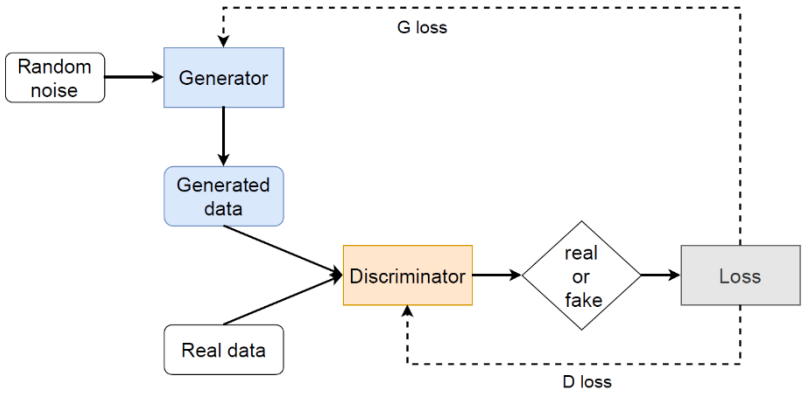


Figure 1. Une représentation schématique de l'architecture d'un réseau antagoniste génératif (GAN).

Pour construire un réseau générateur, nous nous appuyons sur le modèle PointNet, qui est une nouvelle architecture de réseau neuronal pour traiter les données de nuages de points proposée par Qi et al. [93]. PointNet est apparu comme l'une des premières approches à utiliser l'apprentissage profond pour traiter les nuages de points d'origine. Il a révolutionné le domaine en capturant à la fois des caractéristiques locales et globales sans recourir à des représentations intermédiaires telles que les voxels ou les maillages. La structure de PointNet comprend des perceptrons multicouches (MLP) et des fonctions de max pooling. Pour le réseau

discriminateur, nous appliquons les réseaux neuronaux convolutionnels (CNN). L'entrée comprend deux ensembles (c'est-à-dire des nuages de points et une segmentation de données de référence), des nuages de points et des données générées à partir du modèle générateur. Nous alimentons les données d'entrée à travers des couches de convolution 1D, une couche de max pooling, une couche de normalisation de lots et une fonction d'activation Leaky ReLU. Enfin, une couche entièrement connectée est utilisée pour la sortie du score de vraisemblance.

Les données de nuage de points utilisées dans notre étude pour la segmentation du squelette du nouveau-né sont issues d'une base de données de tomographie informatisée contenant 124 sujets. Chaque sujet subit une segmentation manuelle de l'ensemble du squelette à l'aide du logiciel. Ensuite, nous partitionnons le squelette en 23 sections. Pour améliorer la qualité des données à des fins d'entraînement, nous les augmentons en appliquant des algorithmes de rotation et de rééchantillonnage sur les points originaux. En conséquence de cette augmentation, l'ensemble de données comprend 3808 échantillons.

Nous divisons la base de données en 70% pour l'ensemble d'apprentissage, 10% pour l'ensemble de test et le reste pour l'ensemble de validation. Chaque sujet comporte 2048 nuages de points 3D. En ce qui concerne les hyperparamètres du modèle, l'optimisation Adam est utilisée à la fois pour le discriminateur et le générateur avec un beta de 0.5. Le taux d'apprentissage du réseau générateur commence à 0.001, puis il est divisé par 10 toutes les 50 itérations. Dans le discriminateur, le taux d'apprentissage est fixé à 0.001 pour toutes les boucles. Nous entraînons le GAN avec différentes tailles de lots: 1, 2, 8, 16, 32 et 64.

Le meilleur modèle GAN était avec une taille de lot de 1, bouclée sur 150 étapes, qui a retourné un IoU moyen de 93.68% ± 7.37%, un Dice moyen de 96.56% ± 4.41% et un score de précision moyen de 96.72% ± 3.56% pour la segmentation prévue sur l'ensemble de validation. Avec le meilleur algorithme GAN, nous avons obtenu la meilleure segmentation des parties du squelette du nouveau-né prédit avec respectivement 99.63%, 99.81% et 99.98% pour les moyennes de IoU, Dice et score de précision. Pendant ce temps, le pire cas a rendu un IoU moyen de 65.81%, un Dice moyen de 79.37% et un score de précision moyen de 80.03%. La Figure 2 illustre le meilleur cas de segmentation des parties osseuses à partir de nuages de points 3D en utilisant notre modèle GAN.

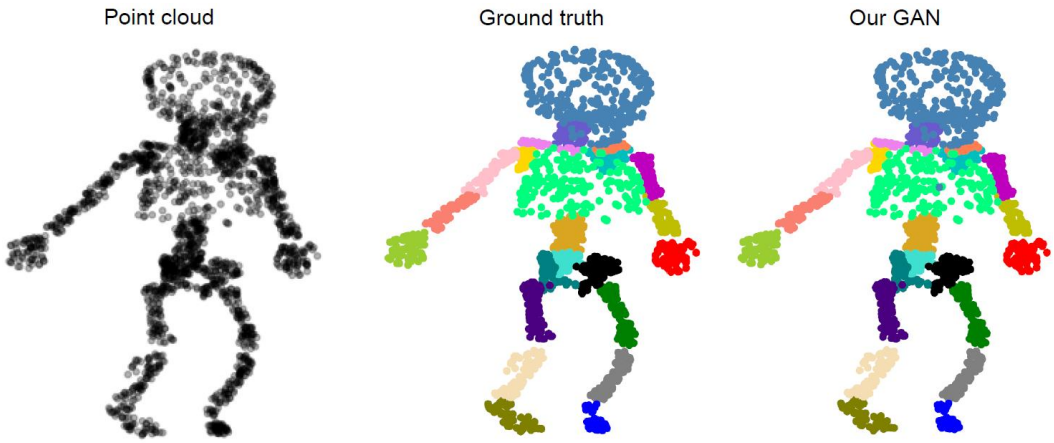


Figure 2. La meilleure segmentation des parties du squelette prédite par notre méthode GAN.

Notre méthode a été comparée au réseau de neurones convolutifs par points (Pointwise CNN) [142]. Les résultats segmentés sur l'ensemble de validation ont obtenu un IoU moyen de 72.30% ± 5.1%, un Dice moyen de 83.82% ± 3.44% et un score de précision moyen de 84.81%

± 3.25%. Le meilleur cas a donné un IoU moyen de 86.31%, un Dice moyen de 88.75% et un score de précision moyen de 89.37%. La prédiction la moins précise a montré un IoU moyen de 45.10%, un Dice moyen de 62.15% et un score de précision de 62.88%.

Dans le chapitre 3, nous avons proposé un réseau antagoniste génératif basé sur PointNet pour segmenter avec précision et rapidité les composants osseux, en seulement 5 secondes. Notre modèle GAN a montré que les performances de segmentation sont supérieures à celles du réseau de neurones convolutifs par points. Cependant, la précision des résultats dépend des postures fœtales. Il reste difficile de séparer les os fusionnés. Par conséquent, les études futures devraient se concentrer sur la segmentation des os fusionnés comme les vertèbres et fusionner l'ensemble du squelette articulé dans le modèle pelvien maternel pour effectuer des simulations d'accouchement complexes et des actions préventives associées.

3. Une nouvelle approche basée sur l'apprentissage profond pour prédire les déformations des tissus mous du bassin vers une simulation d'accouchement interactive en temps réel

La dynamique des tissus mous est cruciale pour comprendre le comportement mécanique du corps humain. Actuellement, des méthodes numériques utilisant la modélisation par éléments finis et des cadres de type masse-ressort sont employées pour réaliser une approximation des propriétés dynamiques des tissus mous biologiques. Malgré leur utilité, ces approches sont handicapées par des dépenses computationnelles significatives, principalement dues à la nécessité de configurations de maillage dans la formulation des équations d'équilibre dynamique et aux défis liés à la convergence instable. Il est essentiel de résoudre ces problèmes pour faire progresser notre compréhension de la dynamique des tissus mous et améliorer l'efficacité des simulations computationnelles dans les études biomécaniques.

Pour estimer la déformation des tissus mous en temps réel et dans des conditions stables, un nouveau flux de travail a été développé et présenté dans la Figure 3. Tout d'abord, une base de données basée sur la simulation a été générée à partir d'une simulation d'accouchement réalisée à l'aide de notre modèle HyperMSM récemment développé [159]. Ensuite, différentes architectures et stratégies d'apprentissage profond ont été conçues, implémentées et évaluées. Enfin, le meilleur modèle d'apprentissage automatique entraîné sera intégré dans une simulation interactive en temps réel de la procédure d'accouchement physiologique.

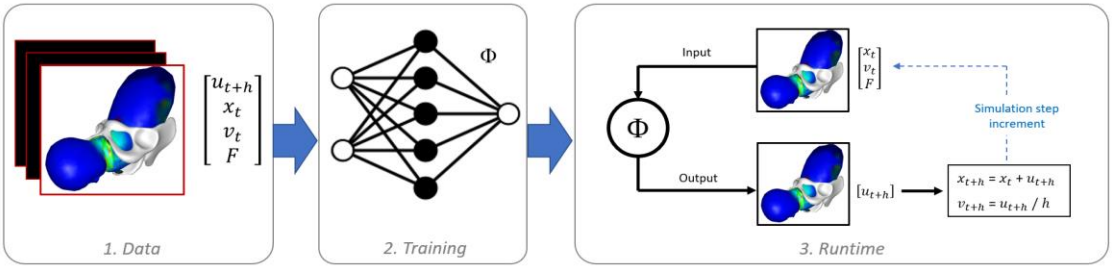


Figure 3. Le flux de travail proposé basé sur les données couple la génération de bases de données, la mise en œuvre et le déploiement d'un réseau d'apprentissage profond.

Nous générons la base de données d'apprentissage en simulant les processus d'accouchement. Notre méthode d'entraînement consiste à prendre des données brutes en séries temporelles, comprenant les positions des sommets, les vitesses et les forces interactives appliquées aux

nœuds, et à prédire les déplacements nodaux. Le modèle comprend un corps fœtal, les tissus mous du bassin tels que les muscles du plancher pelvien, le vagin et l'utérus, ainsi que la structure osseuse du bassin. Nous utilisons HyperMSM pour exécuter de nombreuses simulations du processus d'accouchement tout en conservant les mêmes caractéristiques matérielles de Neo-Hookean. Pour ces simulations, l'intervalle de temps est fixé à 2 ms dans chaque cas. Nous considérons simplement des points de surface non contraints en 3D. Le modèle se compose de 1653 nœuds de surface, ce qui donne des vecteurs de taille 3×1653. Pendant chaque simulation, des forces externes aléatoires sont appliquées à la surface des muscles du plancher pelvien pour simuler des forces d'intervention clinique potentielles. Nous obtenons une base de données d'entraînement de 15396 trames concernant 8 simulations, et une base de données de test de 18633 trames concernant 6 simulations.

Nous avons utilisé une variante des réseaux neuronaux récurrents appelée réseau mémoire à court terme (LSTM) [35]. Le fonctionnement du LSTM peut être décrit en quatre étapes. Initialement, la phase d'oubli se produit, pendant laquelle le LSTM évalue quelles informations de son état de cellule précédent doivent être oubliées. Ensuite, dans la phase de stockage, il détermine quelles nouvelles données seront incorporées dans l'état de la cellule. L'étape suivante consiste à mettre à jour l'état de la cellule précédent pour former le nouveau en intégrant les valeurs obtenues de la phase précédente. Enfin, la phase de sortie, qui implique de déterminer l'influence des informations sur la sortie via une couche sigmoïde, permet au réseau LSTM de saisir les dépendances à long terme. Nous avons également utilisé une variante du LSTM appelée mémoire à court terme (BiLSTM) [97]. Le réseau a été introduit pour relever le défi de capturer les dépendances à long terme dans les données séquentielles. Dans le modèle BiLSTM, l'entrée est fournie à la fois des directions de gauche à droite et de droite à gauche simultanément. En raison de la grande dimensionalité des données d'entrée, nous avons choisi d'appliquer une technique de réduction de modèle pour étudier l'impact de la réduction de modèle sur la précision d'un modèle d'apprentissage en profondeur. L'analyse en composantes principales (ACP) est une méthode statistique pour traduire un groupe de variables corrélées en un groupe de variables indépendantes. Elle est souvent utilisée dans l'analyse des données et le développement de modèles prédictifs en apprentissage automatique.

Nous formons le réseau à travers un processus de réglage d'hyperparamètres impliquant la variation du nombre de couches cachées (de 1 à 3) et de leurs nœuds associés (de 100 à 1000). De plus, des ajustements ont été apportés à la taille du lot (de 10 à 100), aux fonctions d'activation (Linéaire, Sigmoïde, Tanh, ReLU) et au taux d'apprentissage (de 10^{-3} à 10^{-6}). Ce processus de réglage a été réalisé en utilisant Optuna, un cadre d'optimisation des hyperparamètres, avec l'optimisation Adam. Avant de commencer le processus d'apprentissage, les données ont été mises à l'échelle en utilisant des valeurs dans l'intervalle [-1,1] pour normaliser. Pour comparer avec l'architecture LSTM sur la prédiction de la déformation des tissus mous du bassin, un réseau neuronal profond classique (DNN) a été utilisé.

En ce qui concerne l'évaluation sur la base de données de test, l'erreur quadratique moyenne (RMSE) était de 1.062 mm, 0.988 mm et 0.936 mm respectivement pour les modèles LSTM, BiLSTM et DNN. Un coefficient de corrélation de Pearson (PCC) de 0.994 a été obtenu pour les réseaux LSTM et BiLSTM. Un PCC de 0.996 a été atteint avec le réseau DNN. Plus précisément, le meilleur cas prédit à partir de la base de données de test a produit des erreurs moyennes de 0.103 mm ± 0.116 mm et 0.107 mm ± 0.083 mm avec respectivement les réseaux LSTM et DNN. La Figure 4 compare la meilleure déformation prédite en utilisant LSTM à

l'estimation HyperMSM de la simulation d'accouchement aux pas de temps 1s, 5s, 8s et 11.186s lorsque l'erreur absolue maximale de 3.264 mm a été obtenue. Avec le modèle DNN, la meilleure prédiction a renvoyé une erreur absolue maximale de 2.765 mm au saut de temps 11.092s. En ce qui concerne le pire cas prédit, la prédiction à l'aide de LSTM a donné une erreur moyenne absolue de 0.845 mm ± 1.214 mm, tandis que la prévision de déformation basée sur DNN a une erreur moyenne de 0.775 mm ± 1.041 mm.

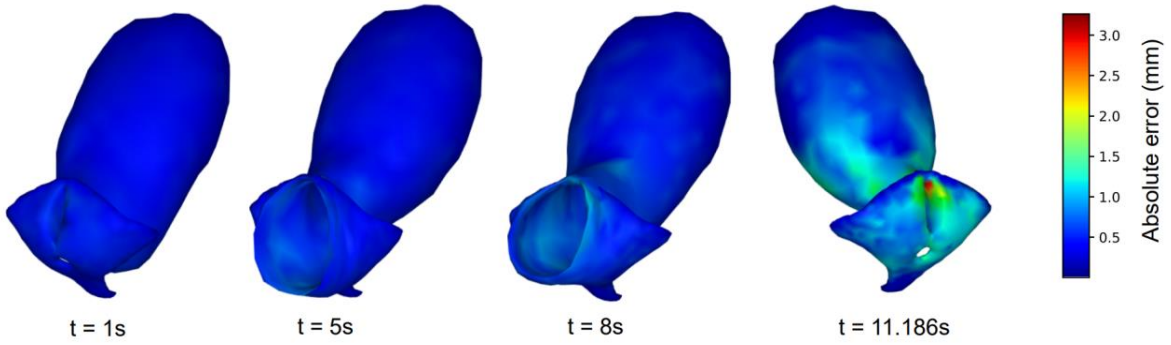


Figure 4. La meilleure simulation prédite par LSTM au fil du temps.

En application sur notre base de données, un nombre optimal de 11 composantes principales a été obtenu pour maximiser la variance des données jusqu'à 99.99% tout en conservant autant de données que possible. Les résultats prédits du déplacement nodal par la stratégie d'apprentissage basée sur l'ACP ont obtenu une RMSE de 0.708 mm et une moyenne ± écart-type de 0.438 ± 0.556 mm pour l'ensemble de données d'entraînement. Les résultats prévus sur les données de test ont renvoyé une RMSE de 1.535 mm à 1.547 mm et une moyenne ± SD de 0.865 ± 1.280 mm.

En utilisant nos modèles d'apprentissage en profondeur sans maillage, nous avons estimé avec précision et rapidement la déformation des tissus mous du bassin lors de l'accouchement du corps fœtal dans le chapitre 4. Les résultats prévus sont bien considérés dans la littérature scientifique. Une fois que le modèle est entraîné, les prédictions peuvent être générées en quelques secondes sur un ordinateur personnel. L'étude actuelle est limitée par son focus uniquement sur des scénarios de délivrance fœtale normaux, en négligeant des situations plus complexes. De plus, les simplifications dans notre modèle d'accouchement, y compris la représentation géométrique et la manipulation des interactions, sont reconnues. Bien que ces simplifications correspondent aux objectifs actuels, des études futures pourraient améliorer le réalisme avec des modèles anatomiquement précis et une simulation de contact avancée.

4. Équations différentielles ordinaires neurales informées par la physique et cadre de modélisation du système masse-ressort pour prédire les déformations des tissus mous en temps réel

Approfondissant la recherche menée dans le chapitre 4, le chapitre 5 s'efforce de faire progresser la simulation de l'accouchement en simulant en temps réel les déformations des tissus mous. Les méthodes numériques traditionnelles, basées sur des configurations de maillage, sont confrontées à des coûts de traitement élevés et à une instabilité dans la convergence. Pour relever ces défis, nous nous tournons vers les réseaux neuronaux informés par la physique (PINN), reconnus pour leur efficacité dans la résolution de problèmes

d’équations différentielles. Notre approche intègre les PINN et les équations différentielles ordinaires neuronales (NeuralODE) avec le cadre de modélisation du système masse-ressort (MSS) pour développer un modèle prédictif de déformation des tissus mous. Nous établissons les bases théoriques de notre méthodologie en utilisant les équations différentielles ordinaires neuronales informées par la physique (PINODE), ainsi que les formulations MSS. Ensuite, nous évaluons l’efficacité de notre approche novatrice à travers des simulations sur des systèmes masse-ressort en 2D et 3D, ainsi que sur une poutre en porte-à-faux en 3D. Enfin, nous appliquons notre modèle pour simuler et évaluer la déformation d’un modèle d’utérus en 3D sous chargement gravitationnel, marquant ainsi une avancée significative dans le développement de simulations d’accouchement réalistes.

Les réseaux neuronaux informés par la physique sont une méthode sans maillage qui apprend directement la physique sous-jacente d’un système à partir des données, et intègre les équations gouvernantes ainsi que les conditions aux limites/initiales dans la fonction de perte (Figure 5). En exploitant cette compréhension physique inhérente comme fondement, les PINN peuvent être entraînés en tant que modèles de substitution pour résoudre des équations différentielles avec un minimum de données étiquetées, voire parfois sans étiquettes du tout. Les PINN exploitent deux caractéristiques fondamentales des réseaux neuronaux. Premièrement, les réseaux neuronaux servent d’approximations de fonctions universelles. Par conséquent, si un réseau neuronal est suffisamment profond et puissant, il peut estimer n’importe quelle fonction, y compris la solution des équations différentielles. Deuxièmement, le calcul des dérivées des sorties d’un réseau neuronal par rapport à ses entrées, ainsi que des paramètres du modèle lors de la rétropropagation, est direct grâce à la différenciation automatique [98].

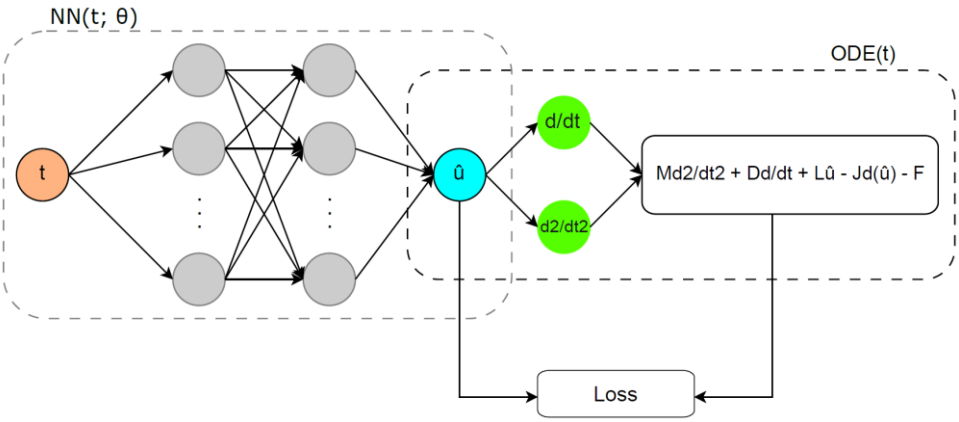


Figure 5. Vue d’ensemble de l’architecture du réseau neuronal informé par la physique.

Nous combinons les réseaux neuronaux informés par la physique et les équations différentielles ordinaires neuronales pour résoudre notre système dynamique lié à la déformation des tissus mous. Le concept fondamental de NeuralODE est lié à la considération des états cachés d’un réseau neuronal comme résultats d’un système d’équations différentielles ordinaires [99]. Dans les réseaux neuronaux conventionnels, la procédure d’apprentissage repose sur l’empilement de couches neuronales et l’application d’une série de transformations paramétrées aux données d’entrée. En revanche, NeuralODE aborde le problème d’un point de vue en temps continu en utilisant des équations différentielles ordinaires pour définir l'évolution des couches cachées d’un réseau neuronal au fil du temps.

Notre approche proposée a été appliquée à l’analyse biomécanique des tissus mous de l’utérus, développée en utilisant le matériau Néo-Hookean avec un module de Young de 0.06 MPa et un

coefficient de Poisson de 0.45. Dans cette simulation, nous avons examiné le déplacement de l'utérus sous la charge gravitationnelle. La représentation structurelle se compose de 1389 nœuds avec 1347 nœuds marqués comme actifs et 42 nœuds fixés du côté gauche. Nous avions un maillage de 4077 éléments tétraédriques. La simulation était caractérisée par 4041 degrés de liberté. Le temps de simulation de l'intervalle de 2 s a été divisé en deux phases. La 1ère seconde est allouée à l'entraînement, et la 2nde seconde implique une extrapolation. La taille du saut de temps est de 0.002 tout au long de la simulation. L'ensemble de données d'entraînement comprenait 500 points. Nous avons appliqué la résolution de formule de différenciation rétrograde d'ODE pour générer les données de référence. La configuration du modèle PINODE se compose d'une seule couche cachée avec 10 nœuds. Le processus d'entraînement implique 4000 itérations avec un taux d'apprentissage de 0.05 pour les 2000 premières itérations et un taux d'apprentissage réduit de 0.01 pour les époques restantes. La fonction d'activation utilisée était la fonction tangente hyperbolique. Pour l'implémentation de PINODE, nous avons utilisé Julia, un langage de programmation haute performance, en utilisant des bibliothèques telles que DiffEqFlux, DifferentialEquations de SciML [228], [229]. Enfin, nous avons appliqué l'optimisation Adam dans toutes les expériences.

La phase d'entraînement du modèle PINODE a donné des résultats prometteurs avec une RMSE de 0.509×10^{-3} m et une erreur relative de norme L^2 de 0.349%. Pendant la phase d'extrapolation, il y a eu une augmentation de la RMSE de 6.404×10^{-3} m et de l'erreur L^2 de 6.357%. La Figure 6 présente une comparaison entre la visualisation de la déformation au temps 2 s en utilisant le modèle PINODE et la vérité terrain correspondante. Une observation intéressante dans cet exemple est que les déplacements extrapolés par PINODE ont été effectués en seulement 1 seconde, tandis que la méthode numérique conventionnelle a nécessité 1 heure.

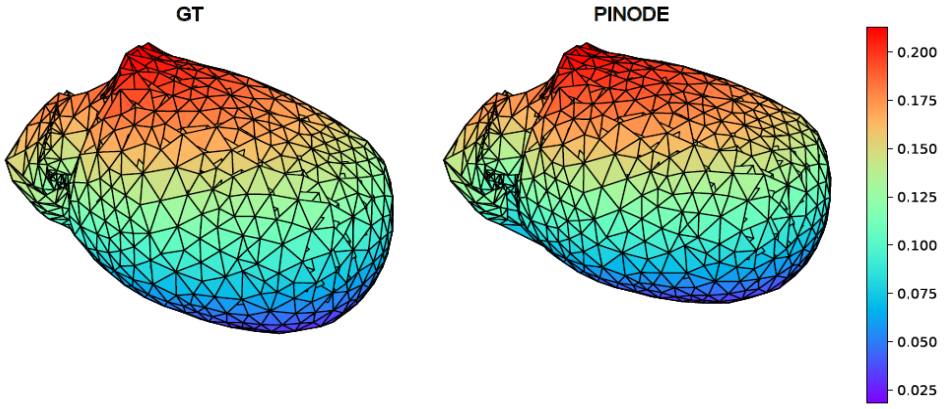


Figure 6. Visualisation de la déformation des tissus mous de l'utérus au temps 2s à l'aide du modèle PINODE par rapport aux données de référence.

Nos modèles PINODE ont montré un fort potentiel dans l'extrapolation des mouvements des systèmes avec de nouvelles conditions initiales par rapport aux modèles PINN classiques. Cependant, le processus d'entraînement de PINODE est difficile en raison de sa structure à profondeur continue. Par conséquent, les modèles PINODE rencontrent également des limitations liées à la convergence et dépendent à la fois de la taille de l'échantillon et des caractéristiques du problème. Les études futures porteraient sur l'investigation de méthodes d'apprentissage fiables pour garantir des prédictions sûres dans la phase d'extrapolation. D'un point de vue applicatif, le modèle actuel des tissus mous de l'utérus est encore limité dans sa représentation des diverses forces impliquées dans le processus de travail et d'accouchement.

Pour obtenir un outil complet de formation à l'accouchement interactif en temps réel, il est nécessaire de disposer d'une base de données plus riche comprenant des scénarios réels et une large gamme de forces, de charges externes et de comportements de contact des tissus mous.

5. Conclusions et perspectives

Cette thèse représente une avancée révolutionnaire dans le domaine de la représentation réaliste du fœtus et de la modélisation en temps réel de la déformation des tissus mous. En intégrant des techniques de pointe et des méthodologies innovantes, elle ouvre de nouvelles voies dans le domaine de la santé, promettant de révolutionner la formation à l'accouchement et la simulation des complications. Avec une attention particulière portée à l'exactitude et à la précision, cette recherche a posé les bases du développement d'un outil d'aide à la décision de nouvelle génération. Ces outils améliorent non seulement l'efficacité de la formation à l'accouchement, mais facilitent également la simulation de diverses complications, fournissant des informations inestimables pour les professionnels de la santé. En comblant le fossé entre la théorie et la pratique, cette thèse marque une avancée significative dans l'amélioration des résultats en matière de santé maternelle et néonatale.

Bien que nos méthodes proposées aient montré des résultats prometteurs, il est important de reconnaître certaines limites. La segmentation des parties du squelette du nouveau-né s'est avérée difficile en raison des os fusionnés, tandis qu'un ensemble de données limité a encore compliqué le processus. Les modèles existants des tissus mous du bassin manquaient d'une représentation complète des forces diverses inhérentes à l'accouchement. Les modèles de simulation de l'accouchement étaient simplistes, se concentrant uniquement sur des circonstances normales et échouant à incorporer les complexités rencontrées dans les cas du monde réel. De plus, pour régir l'interaction entre le corps fœtal et l'utérus maternel, nous avons utilisé des modèles de contact de nœud à triangle simplistes.

Par conséquent, les travaux futurs viseront à aborder ces limites afin de développer un modèle plus complet et précis qui puisse simuler efficacement la dynamique de l'accouchement et offrir des connaissances sur les processus physiologiques impliqués. Dans la simulation de l'accouchement, les composants essentiels comprennent un modèle pelvien maternel et un modèle fœtal détaillé avec des articulations. Les recherches futures visent à intégrer ces modèles cruciaux dans le dispositif Hololens, envisageant une amélioration des compétences en formation à l'accouchement grâce à la création de scénarios d'accouchement physiologiquement plausibles. En fusionnant ces modèles au sein de l'environnement immersif des Hololens, les stagiaires acquerront une précieuse expérience pratique et des perspectives sur les complexités de l'accouchement, favorisant une compréhension plus complète des dynamiques maternelles et fœtales.

References

[1] WHO and UNICEF, "Trends in maternal mortality 2000 to 2020," 2023, [Online]. Available: https://www.who.int/publications/i/item/9789240068759

[2] L. Hug *et al.*, "Global, regional, and national estimates and trends in stillbirths from 2000 to 2019: a systematic assessment," *The Lancet*, vol. 398, no. 10302, pp. 772–785, 2021, doi: https://doi.org/10.1016/S0140-6736(21)01112-0.

[3] UN DESA, UNICEF, and WHO, "A Neglected Tragedy: The global burden of stillbirths - Report of the UN Inter-agency Group for Child Mortality Estimation, 2020," 2020.

[4] H. E. Reinebrant *et al.*, "Making stillbirths visible: a systematic review of globally reported causes of stillbirth," *BJOG*, vol. 125, no. 2, pp. 212–224, 2018, doi: https://doi.org/10.1111/1471-0528.14971.

[5] M. Aminu, R. Unkels, M. Mdegela, B. Utz, S. Adaji, and N. van den Broek, "Causes of and factors associated with stillbirth in low- and middle-income countries: a systematic literature review," *BJOG*, vol. 121, no. s4, pp. 141–153, 2014, doi: https://doi.org/10.1111/1471-0528.12995.

[6] D. L. Hoyert and E. C. W. Gregory, "Cause-of-death data from the fetal death file, 2015–2017," *Natl Vital Stat Rep*, vol. 69, no. 4, pp. 1–20, 2020.

[7] E. S. Draper *et al.*, "MBRRACE-UK Perinatal Mortality Surveillance Report," *UK Perinatal Deaths for Births from January to December*, 2017.

[8] J. E. Lawn *et al.*, "Stillbirths: rates, risk factors, and acceleration towards 2030," *The Lancet*, vol. 387, no. 10018, pp. 587–603, 2016, doi: https://doi.org/10.1016/S0140-6736(15)00837-5.

[9] J. Fahey and T. L. King, "Intrauterine Asphyxia: Clinical Implications for Providers of Intrapartum Care," *J Midwifery Womens Health*, vol. 50, no. 6, pp. 498–506, 2005, doi: https://doi.org/10.1016/j.jmwh.2005.08.007.

[10] E. Jauniaux and D. Jurkovic, "Placenta accreta: Pathogenesis of a 20th century iatrogenic uterine disease," *Placenta*, vol. 33, no. 4, pp. 244–251, 2012, doi: https://doi.org/10.1016/j.placenta.2011.11.010.

[11] S. Behbehani, V. Patenaude, and H. A. Abenhaim, "Maternal Risk Factors and Outcomes of Umbilical Cord Prolapse: A Population-Based Study," *Journal of Obstetrics and Gynaecology Canada*, vol. 38, no. 1, pp. 23–28, 2016, doi: https://doi.org/10.1016/j.jogc.2015.10.008.

[12] U. Korhonen, P. Taipale, and S. Heinonen, "Fetal pelvic index to predict cephalopelvic disproportion – a retrospective clinical cohort study," *Acta Obstet Gynecol Scand*, vol. 94, no. 6, pp. 615–621, 2015, doi: https://doi.org/10.1111/aogs.12608.

[13] P. Frémondière, L. Thollon, P. Adalian, J. Delotte, and F. Marchal, "Which Foetal-Pelvic Variables Are Useful for Predicting Caesarean Section and Instrumental Assistance," *Medical Principles and Practice*, vol. 26, pp. 359–367, 2017, doi: https://doi.org/ https://doi.org/10.1159/000477732

[14] K. Place, H. Kruit, A. Tekay, S. Heinonen, and L. Rahkonen, "Success of trial of labor in women with a history of previous cesarean section for failed labor induction or labor dystocia: A retrospective cohort study," *BMC Pregnancy Childbirth*, vol. 19, Mar. 2019, doi: 10.1186/s12884-019-2334-3.

[15] K. Kawakami, Y. Tanaka, Y. Ikeda, and ..., “Is routine X-ray pelvimetry of value to decide on mode of delivery for women with labor dystocia?,” *Clin Exp Obstet Gynecol*, vol. 48, no. 2, pp. 317–322, 2021.

[16] I. Neuman and A. B. Lalonde, “MamaNatalie: A birthing simulator for realistic training to control postpartum hemorrhage”, [Online]. Available: https://www.glowm.com/pdf/PPH_2nd_edn_Chap-73.pdf

[17] D. C. Jude, G. G. Gilbert, and D. Magrane, “Simulation training in the obstetrics and gynecology clerkship,” *Am J Obstet Gynecol*, vol. 195, no. 5, pp. 1489–1492, 2006, doi: https://doi.org/10.1016/j.ajog.2006.05.003.

[18] Y. Liu, M. Scudder, and M. Gimovsky, “CAD Modelling of the Birth Proces: A Preliminary Report,” University of Massachusetts, USA, 1994.

[19] R. J. Lapeer and R. W. Prager, “Finite Element Model of a Fetal Skull Subjected to Labour Forces,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 1999, doi: https://doi.org/10.1007/10704282_124

[20] R. J. Lapeer and R. W. Prager, “Fetal head moulding: finite element analysis of a fetal skull subjected to uterine pressures during the first stage of labour,” *J Biomech*, vol. 34, no. 9, pp. 1125–1133, 2001, doi: https://doi.org/10.1016/S0021-9290(01)00070-7.

[21] X. Yan, J. Kruger, P. Nielsen, and M. Nash, “Effects of fetal head shape variation on the second stage of labour,” *J Biomech*, vol. 48, Mar. 2015, doi: 10.1016/j.jbiomech.2015.02.062.

[22] L. Krofta, L. Havelková, I. Urbánková, M. Krmář, L. Hyník, and J. Feyereisl, “Finite element model focused on stress distribution in the levator ani muscle during vaginal delivery,” *Int Urogynecol J*, vol. 28, pp. 275–284, 2016, doi: 10.1007/s00192-016-3126-1.

[23] A. Ballit, M. Hivert, C. Rubod, and T.-T. Dao, “Fast soft-tissue deformations coupled with mixed reality toward the next-generation childbirth training simulator,” *Med Biol Eng Comput*, vol. 61, no. 8, pp. 2207–2226, 2023, doi: 10.1007/s11517-023-02864-5.

[24] J.-P. Briot and F. Pachet, “Deep learning for music generation: challenges and directions,” *Neural Comput Appl*, vol. 32, no. 4, pp. 981–993, 2020, doi: 10.1007/s00521-018-3813-6.

[25] A. Esteva *et al.*, “A guide to deep learning in healthcare,” *Nat Med*, vol. 25, no. 1, pp. 24–29, 2019, doi: 10.1038/s41591-018-0316-z.

[26] J. R. Hernandez-Olivan Carlos and Beltrán, “Music Composition with Deep Learning: A Review,” in *Advances in Speech and Music Technology: Computational Aspects and Applications*, E. and W. A. and L. R. H. Biswas Anupam and Wennekes, Ed., Cham: Springer International Publishing, 2023, pp. 25–50. doi: 10.1007/978-3-031-18444-4_2.

[27] A. Motwani, P. K. Shukla, and M. Pawar, “Ubiquitous and smart healthcare monitoring frameworks based on machine learning: A comprehensive review,” *Artif Intell Med*, vol. 134, p. 102431, 2022, doi: https://doi.org/10.1016/j.artmed.2022.102431.

[28] M. A. Rendón, C. D. Sánchez R., J. Gallo M., and A. H. Anzai, “Aircraft Hybrid-Electric Propulsion: Development Trends, Challenges and Opportunities,” *Journal of Control, Automation and Electrical Systems*, vol. 32, no. 5, pp. 1244–1268, 2021, doi: 10.1007/s40313-021-00740-x.

[29] A. I. Torre-Bastida, J. Del Ser, I. Laña, M. Ilardia, M. N. Bilbao, and S. Campos-Cordobés, “Big Data for transportation and mobility: recent advances, trends and challenges,” *IET Intelligent*

Transport Systems, vol. 12, no. 8, pp. 742–755, 2018, doi: https://doi.org/10.1049/iet-its.2018.5188.

[30] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2121–2159, Jul. 2011.

[31] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *ArXiv*, vol. abs/1212.5701, 2012, doi: 10.48550/arXiv.1212.5701.

[32] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, pp. 26-31., 2012.

[33] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, 2014, doi: 10.48550/arXiv.1412.6980.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[35] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[36] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *arXiv:1406.1078*, 2014, doi: 10.48550/arXiv.1406.1078.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *In Proceedings of the IEEE conference on computer vision and pattern recognition* , pp. 770–778, 2016.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer International Publishing, 2015, pp. 234–241.

[39] C. Ye *et al.*, "Prediction of Incident Hypertension Within the Next Year: Prospective Study Using Statewide Electronic Health Records and Machine Learning," *J Med Internet Res*, vol. 20, 2018, doi: 10.2196/jmir.9268.

[40] G. H. Tison *et al.*, "Passive Detection of Atrial Fibrillation Using a Commercially Available Smartwatch," *JAMA Cardiol*, vol. 3, pp. 409–416, 2018, doi: 10.1001/jamacardio.2018.0136.

[41] C. Yang, B. D. Ojha, N. D. Aranoff, P. Green, and N. Tavassolian, "Classification of aortic stenosis using conventional machine learning and deep learning methods based on multi-dimensional cardio-mechanical signals," *Sci Rep*, vol. 10, no. 1, p. 17521, 2020, doi: 10.1038/s41598-020-74519-6.

[42] V. Abedi *et al.*, "Prediction of Long-Term Stroke Recurrence Using Machine Learning Models," *J Clin Med*, vol. 10, no. 6, 2021, doi: 10.3390/jcm10061286.

[43] R. Rava *et al.*, "Automated Collateral Flow Assessment in Acute Ischemic Stroke Patients using Computed Tomography with Artificial Intelligence Algorithms," *World Neurosurg*, vol. 155, Mar. 2021, doi: 10.1016/j.wneu.2021.08.136.

[44] J. Carreras, N. Nakamura, and R. Hamoudi, "Artificial intelligence analysis of gene expression predicted the overall survival of mantle cell lymphoma and a large pan-cancer series," *Healthcare (Basel)*, vol. 10, no. 1, p. 155, 2022, doi: 10.3390/healthcare10010155.

[45] S. El Hussein, P. Chen, L. J. Medeiros, J. Wu, and J. D. Khoury, "Artificial Intelligence-Assisted Mapping of Proliferation Centers in Chronic Lymphocytic Leukemia/ Small Lymphocytic Lymphoma Identifies Patterns That Reliably Distinguish Accelerated Phase and Large Cell Transformation," *Blood*, vol. 138, no. Supplement 1, p. 1558, Nov. 2021, doi: 10.1182/blood-2021-146327.

[46] A. S. AlAgha, H. Faris, B. H. Hammo, and A. M. Al-Zoubi, "Identifying β-thalassemia carriers using a data mining approach: The case of the Gaza Strip, Palestine," *Artif Intell Med*, vol. 88, pp. 70–83, 2018, doi: https://doi.org/10.1016/j.artmed.2018.04.009.

[47] T. A. Retson, E. M. Masutani, D. Golden, and A. Hsiao, "Clinical Performance and Role of Expert Supervision of Deep Learning for Cardiac Ventricular Volumetry: A Validation Study," *Radiol Artif Intell*, vol. 2, no. 4, p. e190064, 2020, doi: 10.1148/ryai.2020190064.

[48] R. Karim *et al.*, "Algorithms for left atrial wall segmentation and thickness – Evaluation on an open-source CT and MRI image database," *Med Image Anal*, vol. 50, pp. 36–53, 2018, doi: https://doi.org/10.1016/j.media.2018.08.004.

[49] X. Tang *et al.*, "Whole liver segmentation based on deep learning and manual adjustment for clinical use in SIRT," *Eur J Nucl Med Mol Imaging*, vol. 47, no. 12, pp. 2742–2752, 2020, doi: 10.1007/s00259-020-04800-3.

[50] N. Bouteldja *et al.*, "Deep Learning–Based Segmentation and Quantification in Experimental Kidney Histopathology," *Journal of the American Society of Nephrology*, vol. 32, no. 1, 2021, doi: 10.1681/ASN.2020050597.

[51] F. Zabihollahy, A. N. Viswanathan, E. J. Schmidt, M. Morcos, and J. Lee, "Fully automated multiorgan segmentation of female pelvic magnetic resonance images with coarse-to-fine convolutional neural network," *Med Phys*, vol. 48, no. 11, pp. 7028–7042, 2021, doi: https://doi.org/10.1002/mp.15268.

[52] J. Lu *et al.*, "Development and application of a detection platform for colorectal cancer tumor sprouting pathological characteristics based on artificial intelligence," *Intelligent Medicine*, vol. 2, no. 2, pp. 82–87, 2022, doi: https://doi.org/10.1016/j.imed.2021.08.003.

[53] A. Rocca *et al.*, "Early Diagnosis of Liver Metastases from Colorectal Cancer through CT Radiomics and Formal Methods: A Pilot Study," *J Clin Med*, vol. 11, no. 1, 2022, doi: 10.3390/jcm11010031.

[54] N. Tomašev *et al.*, "A clinically applicable approach to continuous prediction of future acute kidney injury," *Nature*, vol. 572, no. 7767, pp. 116–119, 2019, doi: 10.1038/s41586-019-1390-1.

[55] Zebra Medical Vision, "Zebra Medical Vision," 2019. [Online]. Available: https://www.accessdata.fda.gov/cdrh_docs/pdf19/K190362.pdf

[56] A. L. Young *et al.*, "Uncovering the heterogeneity and temporal complexity of neurodegenerative diseases with Subtype and Stage Inference," *Nat Commun*, vol. 9, no. 1, p. 4273, 2018, doi: 10.1038/s41467-018-05892-0.

[57] S. Shinde *et al.*, "Predictive markers for Parkinson's disease using deep neural nets on neuromelanin sensitive MRI," *Neuroimage Clin*, vol. 22, p. 101748, 2019, doi: 10.1016/j.nicl.2019.101748.

[58] A. Eshaghi *et al.*, "Identifying multiple sclerosis subtypes using unsupervised machine learning and MRI data," *Nat Commun*, vol. 12, no. 1, p. 2078, 2021, doi: 10.1038/s41467-021-22265-2.

[59] E. Gleichgerrcht *et al.*, “Artificial intelligence for classification of temporal lobe epilepsy with ROI-level MRI data: A worldwide ENIGMA-Epilepsy study,” *Neuroimage Clin*, vol. 31, p. 102765, 2021, doi: https://doi.org/10.1016/j.nicl.2021.102765.

[60] C. Wang and L. Kurgan, “Survey of similarity-based prediction of drug-protein interactions,” *Curr Med Chem*, vol. 27, no. 35, pp. 5856–5886, 2020, doi: 10.2174/0929867326666190808154841.

[61] E. Guney, J. Menche, M. Vidal, and A.-L. Barábasi, “Network-based in silico drug efficacy screening,” *Nat Commun*, vol. 7, no. 1, p. 10331, 2016, doi: 10.1038/ncomms10331.

[62] F. D. Sistare *et al.*, “Towards consensus practices to qualify safety biomarkers for use in early drug development,” *Nat Biotechnol*, vol. 28, no. 5, pp. 446–454, 2010, doi: 10.1038/nbt.1634.

[63] G.-W. Wei, “Protein structure prediction beyond AlphaFold,” *Nat Mach Intell*, vol. 1, no. 8, pp. 336–337, 2019, doi: 10.1038/s42256-019-0086-4.

[64] K. Uziela, D. Menéndez Hurtado, N. Shu, B. Wallner, and A. Elofsson, “ProQ3D: improved model quality assessments using deep learning,” *Bioinformatics*, vol. 33, no. 10, pp. 1578–1580, May 2017, doi: 10.1093/bioinformatics/btw819.

[65] R. Cao, B. Adhikari, D. Bhattacharya, M. Sun, J. Hou, and J. Cheng, “QAcon: single model quality assessment using protein structural and contact information with machine learning techniques,” *Bioinformatics*, vol. 33, no. 4, pp. 586–588, Feb. 2017, doi: 10.1093/bioinformatics/btw694.

[66] R. Cao, D. Bhattacharya, J. Hou, and J. Cheng, “DeepQA: improving the estimation of single protein model quality with deep belief networks,” *BMC Bioinformatics*, vol. 17, no. 1, p. 495, 2016, doi: 10.1186/s12859-016-1405-y.

[67] L. A. Anderson, J. A. Erickson, C. R. Kapron, B. E. Blackburn, and C. L. Peters, “John Charnley Award: A Quantitative Fluoroscopic Tool Improves Acetabular Fragment Positioning in Periacetabular Osteotomy,” *J Arthroplasty*, vol. 38, no. 7, Supplement, pp. S16-S22.e1, 2023, doi: https://doi.org/10.1016/j.arth.2023.03.052.

[68] R. E. Hoyt, D. Snider, C. Thompson, and S. Mantravadi, “IBM Watson Analytics: Automating Visualization, Descriptive, and Predictive Statistics,” *JMIR Public Health Surveill*, vol. 2, no. 2, p. e157, 2016, doi: 10.2196/publichealth.5810.

[69] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric Deep Learning: Going beyond Euclidean data,” *IEEE Signal Process Mag*, vol. 34, no. 4, pp. 18–42, 2017, doi: 10.1109/MSP.2017.2693418.

[70] N. N. Vlassis, R. Ma, and W. Sun, “Geometric deep learning for computational mechanics Part I: anisotropic hyperelasticity,” *Comput Methods Appl Mech Eng*, vol. 371, p. 113299, 2020, doi: https://doi.org/10.1016/j.cma.2020.113299.

[71] I. Goodfellow *et al.*, “Generative Adversarial Networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, doi: 10.1145/3422622.

[72] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv:1411.1784*, 2014, doi: 10.48550/arXiv.1411.1784.

[73] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv:1511.06434*, 2015, doi: 10.48550/arXiv.1511.06434.

[74] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251. doi: 10.1109/ICCV.2017.244.

[75] D. Nie, L. Wang, Y. Gao, J. Lian, and D. Shen, “STRAINet: Spatially Varying sTochastic Residual AdversarIal Networks for MRI Pelvic Organ Segmentation,” *IEEE Trans Neural Netw Learn Syst*, vol. 30, no. 5, pp. 1552–1564, 2019, doi: 10.1109/TNNLS.2018.2870182.

[76] R. Kalantar *et al.*, “CT-Based Pelvic T1-Weighted MR Image Synthesis Using UNet, UNet++ and Cycle-Consistent Generative Adversarial Network (Cycle-GAN),” *Front Oncol*, vol. 11, p. 665807, Apr. 2021, doi: 10.3389/fonc.2021.665807.

[77] G. Li, J. Lv, X. Tong, C. Wang, and G. Yang, “High-Resolution Pelvic MRI Reconstruction Using a Generative Adversarial Network With Attention and Cyclic Loss,” *IEEE Access*, vol. 9, pp. 105951–105964, 2021, doi: 10.1109/ACCESS.2021.3099695.

[78] N. Tong, S. Gou, S. Yang, M. Cao, and K. Sheng, “Shape constrained fully convolutional DenseNet with adversarial training for multiorgan segmentation on head and neck CT and low-field MR images,” *Med Phys*, vol. 46, no. 6, pp. 2669–2682, 2019, doi: 10.1002/mp.13553.

[79] D.-P. Nguyen, M.-C. Ho Ba Tho, and T.-T. Dao, “Reinforcement learning coupled with finite element modeling for facial motion learning,” *Comput Methods Programs Biomed*, vol. 221, p. 106904, 2022, doi: https://doi.org/10.1016/j.cmpb.2022.106904.

[80] D.-P. Nguyen, M.-C. Ho Ba Tho, and T.-T. Dao, “Enhanced facial expression recognition using 3D point sets and geometric deep learning,” *Med Biol Eng Comput*, vol. 59, no. 6, pp. 1235–1244, 2021, doi: 10.1007/s11517-021-02383-1.

[81] G. Yolcu, I. Oztel, S. Kazan, C. Oz, and F. Bunyak, “Deep learning-based face analysis system for monitoring customer interest,” *J Ambient Intell Humaniz Comput*, vol. 11, no. 1, pp. 237–248, 2020, doi: 10.1007/s12652-019-01310-5.

[82] S. Kim, H. Yoon, J. Lee, and S. Yoo, “Facial wrinkle segmentation using weighted deep supervision and semi-automatic labeling,” *Artif Intell Med*, vol. 145, p. 102679, 2023, doi: https://doi.org/10.1016/j.artmed.2023.102679.

[83] S. Lindgren Belal *et al.*, “Deep learning for segmentation of 49 selected bones in CT scans: First step in automated PET/CT-based 3D quantification of skeletal metastases,” *Eur J Radiol*, vol. 113, pp. 89–95, 2019, doi: https://doi.org/10.1016/j.ejrad.2019.01.028.

[84] E. Schnider *et al.*, “Improved distinct bone segmentation in upper-body CT through multi-resolution networks,” *Int J Comput Assist Radiol Surg*, vol. 18, no. 11, pp. 2091–2099, 2023, doi: 10.1007/s11548-023-02957-4.

[85] Z. Cui *et al.*, “A fully automatic AI system for tooth and alveolar bone segmentation from cone-beam CT images,” *Nat Commun*, vol. 13, no. 1, p. 2096, 2022, doi: 10.1038/s41467-022-29637-2.

[86] Y. Zeng, P.-H. Tsui, W. Wu, Z. Zhou, and S. Wu, “Fetal Ultrasound Image Segmentation for Automatic Head Circumference Biometry Using Deeply Supervised Attention-Gated V-Net,” *J Digit Imaging*, vol. 34, no. 1, pp. 134–148, 2021, doi: 10.1007/s10278-020-00410-5.

[87] K. D. McCay, E. S. L. Ho, H. P. H. Shum, G. Fehringer, C. Marcroft, and N. D. Embleton, “Abnormal Infant Movements Classification With Deep Learning on Pose-Based Features,” *IEEE Access*, vol. 8, pp. 51582–51592, 2020, doi: 10.1109/ACCESS.2020.2980269.

[88] P. Agarwal and B. Prabhakaran, “Robust Blind Watermarking of Point-Sampled Geometry,” *Information Forensics and Security, IEEE Transactions on*, vol. 4, pp. 36–48, May 2009, doi: 10.1109/TIFS.2008.2011081.

[89] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view Convolutional Neural Networks for 3D Shape Recognition,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA: IEEE Computer Society, Dec. 2015, pp. 945–953. doi: 10.1109/ICCV.2015.114.

[90] X. Wei, R. Yu, and J. Sun, “Learning View-Based Graph Convolutional Network for Multi-View 3D Shape Analysis,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 6, pp. 7525–7541, 2023, doi: 10.1109/TPAMI.2022.3221785.

[91] S. Zhi, Y. Liu, X. Li, and Y. Guo, “Toward real-time 3D object recognition: A lightweight volumetric CNN framework using multitask learning,” *Comput Graph*, vol. 71, pp. 199–207, 2018, doi: https://doi.org/10.1016/j.cag.2017.10.007.

[92] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, and J. Li, “NormalNet: A voxel-based CNN for 3D object classification and retrieval,” *Neurocomputing*, vol. 323, pp. 139–147, 2019, doi: https://doi.org/10.1016/j.neucom.2018.09.075.

[93] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pp. 77–85, 2017, doi: 10.1109/CVPR.2017.16.

[94] I. Santesteban, E. Garces, M. A. Otaduy, and D. Casas, “SoftSMPL: Data-driven Modeling of Nonlinear Soft-tissue Dynamics for Parametric Humans,” *Computer Graphics Forum*, vol. 39, no. 2, pp. 65–75, 2020, doi: https://doi.org/10.1111/cgf.13912.

[95] L. V. Romaguera, R. Plantefève, F. P. Romero, F. Hébert, J.-F. Carrier, and S. Kadoury, “Prediction of in-plane organ deformation during free-breathing radiotherapy via discriminative spatial transformer networks,” *Med Image Anal*, vol. 64, p. 101754, 2020, doi: https://doi.org/10.1016/j.media.2020.101754.

[96] M. Sacks, S. Motiwale, C. Goodbrake, and W. Zhang, “Neural Network Approaches for Soft Biological Tissue and Organ Simulations,” *J Biomech Eng*, vol. 144, Apr. 2022, doi: 10.1115/1.4055835.

[97] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[98] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, “Automatic differentiation in machine learning: a survey,” *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 5595–5637, Jan. 2017.

[99] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural Ordinary Differential Equations,” *CoRR*, vol. abs/1806.07366, 2018, [Online]. Available: http://arxiv.org/abs/1806.07366

[100] I. T. Jolliffe, “Principal Components in Regression Analysis,” in *Principal Component Analysis*, New York, NY: Springer New York, 1986, pp. 129–155. doi: 10.1007/978-1-4757-1904-8_8.

[101] W. S. Burton, C. A. Myers, and P. J. Rullkoetter, “Machine learning for rapid estimation of lower extremity muscle and joint loading during activities of daily living,” *J Biomech*, vol. 123, p. 110439, 2021, doi: https://doi.org/10.1016/j.jbiomech.2021.110439.

[102] S. Said, Z. Yang, P. Clauser, N. V Ruiter, P. A. T. Baltzer, and T. Hopp, “Estimation of the biomechanical mammographic deformation of the breast using machine learning models,”

Clinical Biomechanics, vol. 110, p. 106117, 2023, doi: https://doi.org/10.1016/j.clinbiomech.2023.106117.

[103] L. Zhang *et al.*, "Spatio-Temporal Convolutional LSTMs for Tumor Growth Prediction by Learning 4D Longitudinal Patient Data," *IEEE Trans Med Imaging*, vol. 39, no. 4, pp. 1114–1126, 2020, doi: 10.1109/TMI.2019.2943841.

[104] E. Lombardo *et al.*, "Evaluation of real-time tumor contour prediction using LSTM networks for MR-guided radiotherapy," *Radiotherapy and Oncology*, vol. 182, p. 109555, 2023, doi: https://doi.org/10.1016/j.radonc.2023.109555.

[105] Q. He, D. Barajas-Solano, G. Tartakovsky, and A. M. Tartakovsky, "Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport," *Adv Water Resour*, vol. 141, p. 103610, 2020, doi: https://doi.org/10.1016/j.advwatres.2020.103610.

[106] S. Guo *et al.*, "Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm," *J Manuf Syst*, vol. 62, pp. 145–163, 2022, doi: https://doi.org/10.1016/j.jmsy.2021.11.003.

[107] 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," *J Comput Phys*, vol. 378, pp. 686–707, 2019, doi: https://doi.org/10.1016/j.jcp.2018.10.045.

[108] Y. Chen, L. Lu, G. E. Karniadakis, and L. D. Negro, "Physics-informed neural networks for inverse problems in nano-optics and metamaterials," *Opt. Express*, vol. 28, no. 8, pp. 11618–11633, Apr. 2020, doi: 10.1364/OE.384875.

[109] R. Zhang, Y. Liu, and H. Sun, "Physics-informed multi-LSTM networks for metamodeling of nonlinear structures," *Comput Methods Appl Mech Eng*, vol. 369, p. 113226, 2020, doi: https://doi.org/10.1016/j.cma.2020.113226.

[110] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, "Physics-Informed Neural Networks for Cardiac Activation Mapping," *Front Phys*, vol. 8, 2020, doi: 10.3389/fphy.2020.00042.

[111] A. Arzani, J.-X. Wang, and R. M. D'Souza, "Uncovering near-wall blood flow from sparse data with physics-informed neural networks," *Physics of Fluids*, vol. 33, no. 7, p. 071905, Jul. 2021, doi: 10.1063/5.0055600.

[112] S. Buoso, T. Joyce, and S. Kozerke, "Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks," *Med Image Anal*, vol. 71, p. 102066, 2021, doi: https://doi.org/10.1016/j.media.2021.102066.

[113] W. Li and K.-M. Lee, "Physics informed neural network for parameter identification and boundary force estimation of compliant and biomechanical systems," *Int J Intell Robot Appl*, vol. 5, no. 3, pp. 313–325, 2021, doi: 10.1007/s41315-021-00196-x.

[114] J. Zhang *et al.*, "Physics-Informed Deep Learning for Musculoskeletal Modeling: Predicting Muscle Forces and Joint Kinematics From Surface EMG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 484–493, 2023, doi: 10.1109/TNSRE.2022.3226860.

[115] M. Movahhedi *et al.*, "Predicting 3D soft tissue dynamics from 2D imaging using physics informed neural networks," *Commun Biol*, vol. 6, no. 1, p. 541, 2023, doi: 10.1038/s42003-023-04914-y.

[116] P. Goyal and P. Benner, “Neural ordinary differential equations with irregular and noisy data,” *R Soc Open Sci*, vol. 10, no. 7, p. 221475, 2023, doi: 10.1098/rsos.221475.

[117] A. Kashyap, S. Plis, P. Ritter, and S. Keilholz, “A deep learning approach to estimating initial conditions of Brain Network Models in reference to measured fMRI data,” *Front Neurosci*, vol. 17, Apr. 2023, doi: 10.3389/fnins.2023.1159914.

[118] Z. Yang *et al.*, “A neural ordinary differential equation model for visualizing deep neural network behaviors in multi-parametric MRI-based glioma segmentation,” *Med Phys*, vol. 50, no. 8, pp. 4825–4838, 2023, doi: https://doi.org/10.1002/mp.16286.

[119] G. Samei, C. Tanner, and G. Székely, “Predicting Liver Motion Using Exemplar Models,” in *Abdominal Imaging. Computational and Clinical Applications*, 2012, pp. 147–157.

[120] C. Tanner, K. Eppenhof, J. Gelderblom, and G. Székely, “Decision fusion for temporal prediction of respiratory liver motion,” in *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, 2014, pp. 698–701. doi: 10.1109/ISBI.2014.6867966.

[121] N. Garau *et al.*, “A ROI-based global motion model established on 4DCT and 2D cine-MRI data for MRI-guidance in radiation therapy,” *Phys Med Biol*, vol. 64, no. 4, p. 45002, Feb. 2019, doi: 10.1088/1361-6560/aafcec.

[122] V. Shah, C. J. Coroneos, and E. Ng, “The evaluation and management of neonatal brachial plexus palsy,” *Paediatr Child Health*, vol. 26, no. 8, pp. 493–497, Dec. 2021, doi: 10.1093/pch/pxab083.

[123] C. R. Macedonia, R. B. Gherman, and A. J. Satin, “Simulation laboratories for training in obstetrics and gynecology,” *Obstetrics & Gynecology*, vol. 102, no. 2, pp. 388–392, 2003, doi: https://doi.org/10.1016/S0029-7844(03)00483-6.

[124] O. Dupuis *et al.*, “Birth simulator: Reliability of transvaginal assessment of fetal head station as defined by the American College of Obstetricians and Gynecologists classification,” *Am J Obstet Gynecol*, vol. 192, no. 3, pp. 868–874, 2005, doi: https://doi.org/10.1016/j.ajog.2004.09.028.

[125] M. P. L. Parente, R. M. N. Jorge, T. Mascarenhas, A. A. Fernandes, and J. A. C. Martins, “The influence of an occipito-posterior malposition on the biomechanical behavior of the pelvic floor,” *European Journal of Obstetrics & Gynecology and Reproductive Biology*, vol. 144, pp. S166–S169, 2009, doi: https://doi.org/10.1016/j.ejogrb.2009.02.033.

[126] S. Chen and M. J. Grimm, “Childbirth Computational Models: Characteristics and Applications,” *J Biomech Eng*, vol. 143, no. 5, p. 50801, May 2021, doi: 10.1115/1.4049226.

[127] R. Lapeer *et al.*, “A computer-based simulation of childbirth using the partial Dirichlet-Neumann contact method with total Lagrangian explicit dynamics on the GPU,” *Biomech Model Mechanobiol*, vol. 18, no. 3, p. 681—700, Jun. 2019, doi: 10.1007/s10237-018-01109-x.

[128] O. Ami *et al.*, “Three-dimensional magnetic resonance imaging of fetal head molding and brain shape changes during the second stage of labor,” *PLoS One*, vol. 14, no. 5, p. e0215721, 2019, doi: 10.1371/journal.pone.0215721.

[129] T. T. Dao, “From deep learning to transfer learning for the prediction of skeletal muscle forces,” *Med Biol Eng Comput*, vol. 57, no. 5, pp. 1049–1058, 2019, doi: 10.1007/s11517-018-1940-y.

[130] A. Ballit and T.-T. Dao, “Recurrent neural network to predict hyperelastic constitutive behaviors of the skeletal muscle,” *Med Biol Eng Comput*, vol. 60, Mar. 2022, doi: 10.1007/s11517-022-02541-z.

[131] D. H. Nguyen-Le, A. Ballit, and T.-T. Dao, “A novel deep learning-driven approach for predicting the pelvis soft-tissue deformations toward a real-time interactive childbirth simulation,” *Eng Appl Artif Intell*, vol. 126, p. 107150, 2023, doi: https://doi.org/10.1016/j.engappai.2023.107150.

[132] N. O’Mahony *et al.*, “Deep Learning vs. Traditional Computer Vision,” in *Advances in Computer Vision*, S. Arai Kohei and Kapoor, Ed., Cham: Springer International Publishing, 2020, pp. 128–144.

[133] Z. Bai and X.-L. Zhang, “Speaker recognition based on deep learning: An overview,” *Neural Networks*, vol. 140, pp. 65–99, 2021, doi: https://doi.org/10.1016/j.neunet.2021.03.004.

[134] B. Norgeot, B. S. Glicksberg, and A. J. Butte, “A call for deep-learning healthcare,” *Nat Med*, vol. 25, no. 1, pp. 14–15, 2019, doi: 10.1038/s41591-018-0320-3.

[135] B. Gürünlü and S. Öztürk, “A Novel Method for Forgery Detection on Lung Cancer Images,” *International Journal of Information Security Science*, vol. 11, no. 3, pp. 13–20, 2022.

[136] P. Liu *et al.*, “Deep learning to segment pelvic bones: large-scale CT datasets and baseline models,” *Int J Comput Assist Radiol Surg*, vol. 16, no. 5, pp. 749–756, 2021, doi: 10.1007/s11548-021-02363-8.

[137] X. Liu *et al.*, “Fully automated pelvic bone segmentation in multiparameteric MRI using a 3D convolutional neural network,” *Insights Imaging*, vol. 12, no. 1, p. 93, 2021, doi: 10.1186/s13244-021-01044-z.

[138] L. Zhang and H. Wang, “A Novel Segmentation Method for Cervical Vertebrae based on PointNet++ and Converge Segmentation,” *Comput Methods Programs Biomed*, vol. 200, p. 105798, Oct. 2020, doi: 10.1016/j.cmpb.2020.105798.

[139] A. Odena, C. Olah, and J. Shlens, “Conditional Image Synthesis with Auxiliary Classifier GANs,” in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., in Proceedings of Machine Learning Research, vol. 70. PMLR, Sep. 2017, pp. 2642–2651. [Online]. Available: https://proceedings.mlr.press/v70/odena17a.html

[140] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2242–2251. doi: 10.1109/ICCV.2017.244.

[141] C. Ma, Y. Yang, J. Guo, F. Pan, C. Wang, and Y. Guo, “Unsupervised Point Cloud Completion and Segmentation by Generative Adversarial Autoencoding Network,” in *Advances in Neural Information Processing Systems*, 2022, pp. 3556–3568. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/171846d7af5ea91e63db508154eaffe8-Paper-Conference.pdf

[142] B. Hua, M. Tran, and S. Yeung, “Pointwise Convolutional Neural Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 984–993. doi: 10.1109/CVPR.2018.00109.

[143] H. Edgar, S. Daneshvari Berry, E. Moes, N. Adolphi, P. Bridges, and K. Nolte, “New Mexico Decedent Image Database (NMDID),” *Office of the Medical Investigator, University of New Mexico*, 2020, doi: 10.25827/5S8C-N515.

[144] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic

segmentation," *Appl Soft Comput*, vol. 70, pp. 41–65, 2018, doi: https://doi.org/10.1016/j.asoc.2018.05.018.

[145] T.-N.-T. Nguyen, A. Ballit, P. Lecomte-Grosbras, J.-B. Colliat, and T.-T. Dao, "On the uncertainty quantification of hyperelastic properties using precise and imprecise probabilities toward reliable in silico simulation of the second-stage labor," *J Mech Med Biol*, p. 2350083, 2023, doi: 10.1142/S0219519423500835.

[146] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning Representations and Generative Models for 3D Point Clouds," in *Proceedings of the 35th International Conference on Machine Learning*, in Proceedings of Machine Learning Research, vol. 80. 2018, pp. 40–49. [Online]. Available: https://proceedings.mlr.press/v80/achlioptas18a.html

[147] Y. Yu, Z. Huang, F. Li, H. Zhang, and X. Le, "Point Encoder GAN: A deep learning model for 3D point cloud inpainting," *Neurocomputing*, vol. 384, pp. 192–199, 2020, doi: https://doi.org/10.1016/j.neucom.2019.12.032.

[148] H. Qin, S. Zhang, Q. Liu, L. Chen, and B. Chen, "PointSkelCNN: Deep Learning-Based 3D Human Skeleton Extraction from Point Clouds," *Computer Graphics Forum*, vol. 39, no. 7, pp. 363–374, 2020, doi: https://doi.org/10.1111/cgf.14151.

[149] A. Takmaz *et al.*, "3D Segmentation of Humans in Point Clouds with Synthetic Data," *arXiv e-prints*, p. arXiv:2212.00786, Dec. 2022, doi: 10.48550/arXiv.2212.00786.

[150] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 5105–5114.

[151] Z. Wang and F. Lu, "VoxSegNet: Volumetric CNNs for Semantic Part Segmentation of 3D Shapes," *IEEE Trans Vis Comput Graph*, vol. 26, pp. 2919–2930, 2018, doi: https://doi.org/10.1109/TVCG.2019.2896310

[152] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, "PartNet: A Recursive Part Decomposition Network for Fine-Grained and Hierarchical Shape Segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2019, pp. 9483–9492. doi: 10.1109/CVPR.2019.00972.

[153] T. T. Dao and M.-C. H. B. Tho, "A Systematic Review of Continuum Modeling of Skeletal Muscles: Current Trends, Limitations, and Recommendations," *Appl Bionics Biomech*, vol. 2018, p. 7631818, 2018, doi: 10.1155/2018/7631818.

[154] T.-N. Nguyen, M.-C. Ho Ba Tho, and T.-T. Dao, "A Systematic Review of Real-Time Medical Simulations with Soft-Tissue Deformation: Computational Approaches, Interaction Devices, System Architectures, and Clinical Validations," *Appl Bionics Biomech*, vol. 2020, p. 5039329, 2020, doi: 10.1155/2020/5039329.

[155] S. Chen and M. Grimm, "Childbirth Computational Models: Characteristics and Applications," *J Biomech Eng*, vol. 143, Dec. 2020, doi: 10.1115/1.4049226.

[156] B. Borotikar, M. Lempereur, M. Lelievre, V. Burdin, D. Ben Salem, and S. Brochard, "Dynamic MRI to quantify musculoskeletal motion: A systematic review of concurrent validity and reliability, and perspectives for evaluation of musculoskeletal disorders," *PLoS One*, vol. 12, p. e0189587, Dec. 2017, doi: 10.1371/journal.pone.0189587.

[157] I. Sack, "Magnetic resonance elastography from fundamental soft-tissue mechanics to diagnostic imaging," *Nature Reviews Physics*, vol. 5, no. 1, pp. 25–42, 2023, doi: 10.1038/s42254-022-00543-2.

[158] M. C. P. Vila Pouca, J. P. S. Ferreira, D. A. Oliveira, M. P. L. Parente, and R. M. Natal Jorge, "Viscous effects in pelvic floor muscles during childbirth: A numerical study," *Int J Numer Method Biomed Eng*, vol. 34, no. 3, p. e2927, Mar. 2018, doi: https://doi.org/10.1002/cnm.2927.

[159] A. Ballit and T.-T. Dao, "HyperMSM: A new MSM variant for efficient simulation of dynamic soft-tissue deformations," *Comput Methods Programs Biomed*, vol. 216, p. 106659, 2022, doi: https://doi.org/10.1016/j.cmpb.2022.106659.

[160] J. Song, H. Xie, Y. Zhong, J. Li, C. Gu, and K.-S. Choi, "Reduced-Order Extended Kalman Filter for Deformable Tissue Simulation," *J Mech Phys Solids*, vol. 158, p. 104696, 2022, doi: https://doi.org/10.1016/j.jmps.2021.104696.

[161] H. Xie, J. Song, Y. Zhong, C. Gu, and K.-S. Choi, "Constrained finite element method for runtime modeling of soft tissue deformation," *Appl Math Model*, vol. 109, pp. 599–612, 2022, doi: https://doi.org/10.1016/j.apm.2022.05.020.

[162] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[163] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[164] H. Fujita, "AI-based computer-aided diagnosis (AI-CAD): the latest review to read first," *Radiol Phys Technol*, vol. 13, no. 1, pp. 6–19, 2020, doi: 10.1007/s12194-019-00552-4.

[165] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, doi: https://doi.org/10.1016/0893-6080(89)90020-8.

[166] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989, doi: https://doi.org/10.1016/0893-6080(89)90003-8.

[167] D. Nguyen, D. Do, J. Lee, T. Rabczuk, and H. Nguyen-Xuan, "Forecasting Damage Mechanics by Deep Learning," *Computers, Materials & Continua*, vol. 61, pp. 951–977, Jan. 2019, doi: 10.32604/cmc.2019.08001.

[168] D. H. Nguyen-Le, Q. B. Tao, V.-H. Nguyen, M. Abdel-Wahab, and H. Nguyen-Xuan, "A data-driven approach based on long short-term memory and hidden Markov model for crack propagation prediction," *Eng Fract Mech*, vol. 235, p. 107085, 2020, doi: https://doi.org/10.1016/j.engfracmech.2020.107085.

[169] H. T. Kollmann, D. W. Abueidda, S. Koric, E. Guleryuz, and N. A. Sobh, "Deep learning for topology optimization of 2D metamaterials," *Mater Des*, vol. 196, p. 109098, 2020, doi: https://doi.org/10.1016/j.matdes.2020.109098.

[170] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, "A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics," *Comput Methods Appl Mech Eng*, vol. 379, p. 113741, 2021, doi: https://doi.org/10.1016/j.cma.2021.113741.

[171] R. Maziar, Y. Alireza, and K. G. Em, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,” *Science (1979)*, vol. 367, no. 6481, pp. 1026–1030, Feb. 2020, doi: 10.1126/science.aaw4741.

[172] P. Garnier, J. Viquerat, J. Rabault, A. Larcher, A. Kuhnle, and E. Hachem, “A review on deep reinforcement learning for fluid mechanics,” *Comput Fluids*, vol. 225, p. 104973, 2021, doi: https://doi.org/10.1016/j.compfluid.2021.104973.

[173] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, “Machine learning for molecular and materials science,” *Nature*, vol. 559, no. 7715, pp. 547–555, 2018, doi: 10.1038/s41586-018-0337-2.

[174] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet – A deep learning architecture for molecules and materials,” *J Chem Phys*, vol. 148, no. 24, p. 241722, Mar. 2018, doi: 10.1063/1.5019779.

[175] K. He, X. Cao, Y. Shi, D. Nie, Y. Gao, and D. Shen, “Pelvic Organ Segmentation Using Distinctive Curve Guided Fully Convolutional Networks.,” *IEEE Trans Med Imaging*, vol. 38 2, pp. 585–595, 2019.

[176] Z. Ma, J. Tavares, R. Natal Jorge, and T. Mascarenhas, “A review of algorithms for medical image segmentation and their applications to the female pelvic cavity,” *Comput Methods Biomech Biomed Engin*, vol. 13, pp. 235–246, Sep. 2009, doi: 10.1080/10255840903131878.

[177] E. Brion, J. Léger, A. M. Barragán-Montero, N. Meert, J. A. Lee, and B. Macq, “Domain adversarial networks and intensity-based data augmentation for male pelvic organ segmentation in cone beam CT,” *Comput Biol Med*, vol. 131, p. 104269, 2021, doi: https://doi.org/10.1016/j.compbiomed.2021.104269.

[178] E. Dandıl and S. Karaca, “Detection of pseudo brain tumors via stacked LSTM neural networks using MR spectroscopy signals,” *Biocybern Biomed Eng*, vol. 41, no. 1, pp. 173–195, 2021, doi: https://doi.org/10.1016/j.bbe.2020.12.003.

[179] A. Zaroug, D. T. H. Lai, K. Mudie, and R. Begg, “Lower Limb Kinematics Trajectory Prediction Using Long Short-Term Memory Neural Networks,” *Front Bioeng Biotechnol*, vol. 8, 2020, doi: 10.3389/fbioe.2020.00362.

[180] S. A. Khadem and A. D. Rey, “Nucleation and growth of cholesteric collagen tactoids: A time-series statistical analysis based on integration of direct numerical simulation (DNS) and long short-term memory recurrent neural network (LSTM-RNN),” *J Colloid Interface Sci*, vol. 582, pp. 859–873, 2021, doi: https://doi.org/10.1016/j.jcis.2020.08.052.

[181] T. Kirchdoerfer and M. Ortiz, “Data-driven computational mechanics,” *Comput Methods Appl Mech Eng*, vol. 304, pp. 81–101, 2016, doi: https://doi.org/10.1016/j.cma.2016.02.001.

[182] C. Mou, B. Koc, O. San, L. G. Rebholz, and T. Iliescu, “Data-driven variational multiscale reduced order models,” *Comput Methods Appl Mech Eng*, vol. 373, p. 113470, 2021, doi: https://doi.org/10.1016/j.cma.2020.113470.

[183] T. Kirchdoerfer and M. Ortiz, “Data-driven computing in dynamics,” *Int J Numer Methods Eng*, vol. 113, no. 11, pp. 1697–1710, 2018, doi: https://doi.org/10.1002/nme.5716.

[184] Q. He, D. W. Laurence, C.-H. Lee, and J.-S. Chen, “Manifold learning based data-driven modeling for soft biological tissues,” *J Biomech*, vol. 117, p. 110124, 2021, doi: https://doi.org/10.1016/j.jbiomech.2020.110124.

[185] V. Tac, V. D. Sree, M. K. Rausch, and A. B. Tepole, “Data-driven modeling of the mechanical behavior of anisotropic soft biological tissue,” *Eng Comput*, vol. 38, no. 5, pp. 4167–4182, 2022, doi: 10.1007/s00366-022-01733-3.

[186] J. Lepage *et al.*, “Biomechanical pregnant pelvic system model and numerical simulation of childbirth: impact of delivery on the uterosacral ligaments, preliminary results,” *Int Urogynecol J*, vol. 26, no. 4, pp. 497–504, 2015, doi: 10.1007/s00192-014-2498-3.

[187] Y. C. Fung, *Biomechanics: Mechanical Properties of Living Tissues*. Springer New York, 1981. [Online]. Available: https://books.google.fr/books?id=HkhRAAAAMAAJ

[188] E. J. Chen, J. Novakofski, W. K. Jenkins, and W. D. O’Brien, “Young’s modulus measurements of soft tissues with application to elasticity imaging,” *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 43, no. 1, pp. 191–194, 1996, doi: 10.1109/58.484478.

[189] R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé, “Biomechanical simulation of the fetal descent without imposed theoretical trajectory,” *Comput Methods Programs Biomed*, vol. 111, no. 2, pp. 389–401, 2013, doi: https://doi.org/10.1016/j.cmpb.2013.04.005.

[190] S. R. Ward and R. L. Lieber, “Density and hydration of fresh and fixed human skeletal muscle,” *J Biomech*, vol. 38, no. 11, pp. 2317–2320, 2005, doi: https://doi.org/10.1016/j.jbiomech.2004.10.001.

[191] S. Chen, M. R. Routzong, S. D. Abramowitch, and M. J. Grimm, “A Computational Procedure to Derive the Curve of Carus for Childbirth Computational Modeling,” *J Biomech Eng*, vol. 145, no. 1, Aug. 2022, doi: 10.1115/1.4055108.

[192] M. J. Grimm, “Forces Involved with Labor and Delivery—A Biomechanical Perspective,” *Ann Biomed Eng*, vol. 49, no. 8, pp. 1819–1835, 2021, doi: 10.1007/s10439-020-02718-3.

[193] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly, “Projective Dynamics: Fusing Constraint Projections for Fast Simulation,” in *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 1st ed., New York, NY, USA: Association for Computing Machinery, 2023. doi: 10.1145/3596711.3596794.

[194] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: continual prediction with LSTM,” in *1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470)*, 1999, pp. 850–855 vol.2. doi: 10.1049/cp:19991218.

[195] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005, doi: https://doi.org/10.1016/j.neunet.2005.06.042.

[196] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” *Neurocomputing*, vol. 337, pp. 325–338, 2019, doi: https://doi.org/10.1016/j.neucom.2019.01.078.

[197] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[198] A. Treuille, A. Lewis, and Z. Popović, “Model Reduction for Real-Time Fluids,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 826–834, Jul. 2006, doi: 10.1145/1141911.1141962.

[199] C. Ooi, Q. T. Le, M. H. Dao, V. B. Nguyen, H.-H. Nguyen, and T. Ba, “Modeling transient fluid simulations with proper orthogonal decomposition and machine learning,” *Int J Numer Methods Fluids*, vol. 93, pp. 396–410, 2020, doi: https://doi.org/10.1002/fld.4888.

[200] S. Armon, E. Efrati, R. Kupferman, and E. Sharon, “Geometry and Mechanics in the Opening of Chiral Seed Pods,” *Science (1979)*, vol. 333, no. 6050, pp. 1726–1730, 2011, doi: 10.1126/science.1203874.

[201] C. Brandt, E. Eisemann, and K. Hildebrandt, “Hyper-Reduced Projective Dynamics,” *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018, doi: 10.1145/3197517.3201387.

[202] D. Holden, B. C. Duong, S. Datta, and D. Nowrouzezahrai, “Subspace Neural Physics: Fast Data-Driven Interactive Simulation,” in *Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, in SCA ’19. New York, NY, USA: Association for Computing Machinery, 2019. doi: 10.1145/3309486.3340245.

[203] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *Optuna: A Next-generation Hyperparameter Optimization Framework*. 2019. doi: 10.1145/3292500.3330701.

[204] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.

[205] J. Benesty, J. Chen, and Y. Huang, “On the Importance of the Pearson Correlation Coefficient in Noise Reduction,” *IEEE Trans Audio Speech Lang Process*, vol. 16, no. 4, pp. 757–765, 2008, doi: 10.1109/TASL.2008.919072.

[206] J. Adler and I. Parmryd, “Quantifying colocalization by correlation: The Pearson correlation coefficient is superior to the Mander’s overlap coefficient,” *Cytometry Part A*, vol. 77A, no. 8, pp. 733–742, 2010, doi: https://doi.org/10.1002/cyto.a.20896.

[207] M. Tonutti, G. Gras, and G.-Z. Yang, “A machine learning approach for real-time modelling of tissue deformation in image-guided neurosurgery,” *Artif Intell Med*, vol. 80, pp. 39–47, Jul. 2017, doi: 10.1016/j.artmed.2017.07.004.

[208] L. Liang, M. Liu, C. Martin, and W. Sun, “A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis,” *J R Soc Interface*, vol. 15, no. 138, p. 20170844, 2018, doi: 10.1098/rsif.2017.0844.

[209] F. Meister, T. Passerini, V. Mihalef, A. Tuysuzoglu, A. Maier, and T. Mansi, “Deep learning acceleration of Total Lagrangian Explicit Dynamics for soft tissue mechanics,” *Comput Methods Appl Mech Eng*, vol. 358, p. 112628, 2020, doi: https://doi.org/10.1016/j.cma.2019.112628.

[210] O. J. Pellicer-Valero, M. J. Rupérez, S. Martínez-Sanchis, and J. D. Martín-Guerrero, “Real-time biomechanical modeling of the liver using Machine Learning models trained on Finite Element Method simulations,” *Expert Syst Appl*, vol. 143, p. 113083, 2020, doi: https://doi.org/10.1016/j.eswa.2019.113083.

[211] M. Pfeiffer, C. Riediger, J. Weitz, and S. Speidel, “Learning soft tissue behavior of organs for surgical navigation with convolutional neural networks,” *Int J Comput Assist Radiol Surg*, vol. 14, no. 7, pp. 1147–1155, 2019, doi: 10.1007/s11548-019-01965-7.

[212] A. Madani, A. Bakhaty, J. Kim, Y. Mubarak, and M. Mofrad, “Bridging Finite Element and Machine Learning Modeling: Stress Prediction of Arterial Walls in Atherosclerosis,” *J Biomech Eng*, vol. 141, Mar. 2019, doi: 10.1115/1.4043290.

[213] A. Mendizabal, E. Tagliabue, J.-N. Brunet, D. Dall’Alba, P. Fiorini, and S. Cotin, “Physics-Based Deep Neural Network for Real-Time Lesion Tracking in Ultrasound-Guided Breast Biopsy,” in *Computational Biomechanics for Medicine*, 2020, pp. 33–45.

[214] G. Joldes *et al.*, "Suite of meshless algorithms for accurate computation of soft tissue deformation for surgical simulation," *Med Image Anal*, vol. 56, pp. 152–171, 2019, doi: https://doi.org/10.1016/j.media.2019.06.004.

[215] L. P. Obrezkov, M. K. Matikainen, and A. B. Harish, "A finite element for soft tissue deformation based on the absolute nodal coordinate formulation," *Acta Mech*, vol. 231, no. 4, pp. 1519–1538, 2020, doi: 10.1007/s00707-019-02607-4.

[216] R. Phellan Aro, B. Hachem, J. Clin, J.-M. Mac-Thiong, and L. Duong, "Real- time biomechanics using the finite element method and machine learning: Review and perspective," *Med Phys*, vol. 48, Dec. 2020, doi: 10.1002/mp.14602.

[217] Y. Zhong, B. Shirinzadeh, G. Alici, and J. Smith, "A Cellular Neural Network Methodology for Deformable Object Simulation," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 749–762, 2006, doi: 10.1109/TITB.2006.875679.

[218] J. Zhang, Y. Zhong, and C. Gu, "Neural network modelling of soft tissue deformation for surgical simulation," *Artif Intell Med*, vol. 97, pp. 61–70, 2019, doi: https://doi.org/10.1016/j.artmed.2018.11.001.

[219] J. R. Mianroodi, N. H. Siboni, and D. Raabe, "Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials," *NPJ Comput Mater*, vol. 7, no. 1, p. 99, 2021, doi: 10.1038/s41524-021-00571-z.

[220] F. Arnold and R. King, "State–space modeling for control based on physics-informed neural networks," *Eng Appl Artif Intell*, vol. 101, p. 104195, 2021, doi: https://doi.org/10.1016/j.engappai.2021.104195.

[221] S. Schaffer *et al.*, "Physics-informed machine learning and stray field computation with application to micromagnetic energy minimization," *J Magn Magn Mater*, vol. 576, p. 170761, 2023, doi: https://doi.org/10.1016/j.jmmm.2023.170761.

[222] J. Luo, F. Abdullah, and P. D. Christofides, "Model predictive control of nonlinear processes using neural ordinary differential equation models," *Comput Chem Eng*, vol. 178, p. 108367, 2023, doi: https://doi.org/10.1016/j.compchemeng.2023.108367.

[223] M. A. Roehrl, T. A. Runkler, V. Brandtstetter, M. Tokic, and S. Obermayer, "Modeling System Dynamics with Physics-Informed Neural Networks Based on Lagrangian Mechanics.," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9195–9200, 2020, doi: https://doi.org/10.1016/j.ifacol.2020.12.2182.

[224] K. Xu and E. Darve, "ADCME: Learning Spatially-varying Physical Fields using Deep Neural Networks," *arXiv:2011.11955*, 2020, doi: 10.48550/arXiv.2011.11955.

[225] O. Hennigh *et al.*, "NVIDIA SimNet™: An AI-Accelerated Multi-Physics Simulation Framework," in *Computational Science – ICCS 2021: 21st International Conference, Krakow, Poland, June 16–18, 2021, Proceedings, Part V*, Berlin, Heidelberg: Springer-Verlag, 2021, pp. 447–461. doi: 10.1007/978-3-030-77977-1_36.

[226] E. Haghighat and R. Juanes, "SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks," *Comput Methods Appl Mech Eng*, vol. 373, p. 113552, 2021, doi: https://doi.org/10.1016/j.cma.2020.113552.

[227] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: A Deep Learning Library for Solving Differential Equations," *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021, doi: 10.1137/19M1274067.

[228] C. Rackauckas, M. Innes, Y. Ma, J. Bettencourt, L. White, and V. Dixit, "DiffEqFlux.jl - A Julia Library for Neural Differential Equations," *arXiv:1902.02376*, 2019, doi: 10.48550/arXiv.1902.02376.

[229] C. Rackauckas *et al.*, "Universal Differential Equations for Scientific Machine Learning," *arXiv:2001.04385*, Feb. 2020, doi: 10.48550/arXiv.2001.04385.

[230] M. Bazmara, M. Silani, M. Mianroodi, and M. Sheibanian, "Physics-informed neural networks for nonlinear bending of 3D functionally graded beam," *Structures*, vol. 49, pp. 152–162, 2023, doi: https://doi.org/10.1016/j.istruc.2023.01.115.

[231] C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," *Comput Methods Appl Mech Eng*, vol. 403, p. 115671, 2023, doi: https://doi.org/10.1016/j.cma.2022.115671.

[232] Z. Lai, C. Mylonas, S. Nagarajaiah, and E. Chatzi, "Structural identification with physics-informed neural ordinary differential equations," *J Sound Vib*, vol. 508, p. 116196, 2021, doi: https://doi.org/10.1016/j.jsv.2021.116196.

[233] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators," *Nat Mach Intell*, vol. 3, no. 3, pp. 218–229, 2021, doi: 10.1038/s42256-021-00302-5.

[234] S. Wang, H. Wang, and P. Perdikaris, "Learning the solution operator of parametric partial differential equations with physics-informed DeepONets," *Sci Adv*, vol. 7, no. 40, p. eabi8605, 2021, doi: 10.1126/sciadv.abi8605.

Titre en français : Modélisation avancée du système d'accouchement à l'aide de différentes méthodes d'apprentissage profond : de la segmentation du squelette fœtal à la déformation des tissus mous en temps réel.

Résumé en français :

Le rapport annuel de l'Organisation mondiale de la santé (OMS) met en évidence une statistique stupéfiante de 100 millions d'accouchements se produisant dans le monde chaque année. Cependant, une réalité sombre accompagne ce chiffre élevé, car le taux de mortalité atteint environ un demi-million. Un facteur contributif significatif à ce taux de mortalité alarmant est le travail obstétrical compliqué. Des simulations d'accouchement ont été étudiées dans le but de prévoir et de prévenir les complications graves chez les mères et les fœtus pendant l'accouchement. La construction d'un modèle complet du système pelvien de la femme enceinte et du corps fœtal est importante pour faire progresser la compréhension des mécanismes de l'accouchement. Cependant, des défis scientifiques persistent dans la représentation réaliste du fœtus et dans le coût computationnel et la vitesse de traitement adaptés pour déployer les simulations d'accouchement dans les pratiques cliniques courantes.

Cette thèse de doctorat apporte trois contributions originales pour surmonter ces défis : 1) la segmentation automatique du squelette fœtal en parties distinctes à l'aide d'un modèle basé sur les réseaux antagonistes génératifs (GAN) et des données de nuage de points 3D ; 2) la prédiction de la déformation des tissus mous en temps réel à l'aide de réseaux neuronaux récurrents (c'est-à-dire des réseaux neuronaux à mémoire à court et long terme (LSTM)) couplés à une stratégie d'apprentissage basée sur l'analyse en composantes principales (ACP) ; et 3) le développement et l'évaluation d'un modèle exceptionnel pour simuler les déformations en temps réel des tissus mous en utilisant les réseaux neuronaux informés par la physique (PINN) et les équations différentielles ordinaires neurales (NeuralODE).

Cette thèse ouvre de nouvelles voies dans la modélisation réaliste de la représentation fœtale et de la déformation en temps réel des tissus mous vers un outil de soutien à la décision de nouvelle génération pour la formation à l'accouchement et la simulation des complications.

Mots-clés : Modélisation prédictive, accouchement, apprentissage profond, systèmes dynamiques, squelette fœtal, tissu mou.

Titre en anglais: Advanced modeling of the childbirth system using different deep learning methods: from fetal skeleton segmentation to real-time soft tissue deformation.

Résumé en anglais:

The annual World Health Organization (WHO) report highlights a staggering statistic of 100 million childbirths occurring worldwide each year. However, a somber reality accompanies this high number, as the death rate reaches approximately half a million. A significant contributing factor to this alarming mortality rate is complicated obstructed labor. Childbirth simulations have been investigated in an attempt to forecast and prevent severe complications in both mothers and fetuses during delivery. Constructing a comprehensive model of the pregnant woman's pelvic system and the fetal body is crucial in advancing the understanding of childbirth mechanics. However, scientific challenges remain in the realistic representation of the fetus and suitable computational cost and processing speed to deploy the childbirth simulations into the clinical routine practices.

This PhD thesis has three original contributions to overcome these challenges: 1) automatic fetal skeleton segmentation into distinct parts using generative adversarial networks (GAN)-based model and 3D point cloud data; 2) prediction of real-time soft tissue deformation using recurrent neural networks (i.e. long short-term memory neural networks (LSTM)) coupled with principal component analysis (PCA)-based learning strategy; and 3) development and evaluation of an outstanding model to simulate real-time deformations of soft tissue using the physics-informed Neural Networks (PINN) and Neural Ordinary Differential Equations (NeuralODE).

This thesis opens new avenues in the realistic modeling of the fetal representation and real-time soft tissue deformation toward a next-generation decision support tool for childbirth training and complication simulation.

Mots-clefs: Predictive modeling, childbirth, deep learning, dynamic systems, fetal skeleton, soft tissue.