Mechanics of Materials 180 (2023) 104631

Contents lists available at ScienceDirect

Mechanics of Materials

journal homepage: www.elsevier.com/locate/mechmat

Machine learning-based prediction for single-cell mechanics

Danh Nguyen[b,1], Lei Tao[a,1], Huilin Ye[a], Ying Li[a,b,*]

[a] Department of Mechanical Engineering, University of Connecticut, Storrs, Connecticut 06269 - 3139, United States
[b] Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, WI 53706 - 1572, United States

ARTICLE INFO

Keywords:
Single-cell mechanics
Capsule deformation
OpenFSI
Machine learning
Convolutional neural network

ABSTRACT

Single-cell mechanics have gained much attention due to its importance in a broad range of biological applications. Different experimental approaches have been used for measuring the mechanical properties of individual cells. However, the technical demands and time-consuming nature of these procedures have limited the throughput of single-cell measurement, necessitating the development of alternative computational approaches. Recently, single-cell deformability can be predicted using a convolutional neural network (CNN) model, shedding a light on using machine learning (ML) algorithms for high-throughput characterizations of single-cell mechanical properties. In this work, we developed a novel ML-based computational framework that can reproduce a physical microfluidic system to investigate the individual cell's deformability. The datasets for the training and testing of our model were generated using high-fidelity fluid-structure interaction (FSI) simulations. Our FSI-based ML approach of adopting CNN algorithms demonstrated a highly accurate prediction for the membrane stiffness of a microcapsule (maximum $R^2 = 0.98$) based on its deformed shape. In this paper, we show that by applying physical constraints including the microcapsule's total surface area and total volume, we were able to build a physics-constrained ML model that possesses better convergence and higher stability during both training and validation. Finally, we found that ML models that used the three-dimensional geometry of the capsule as input could outperform the typical CNN models that relied solely on the two-dimensional images. We expect that this physics-constrained computational framework will serve as a basis for developing future tools for real-time biological applications through the integration of high-fidelity simulations with ML algorithms.

1. Introduction

The mechanical properties of the cell are the behaviors the cell exhibits when physical force is applied to the cell, and play an important role in many biological activities, including cell growth, cell division, cell motility, and adhesion (Hao et al., 2020; Rodriguez et al., 2013). Several subcellular structures are involved in the mechanical behaviors of individual cells, including the cell membrane, cytosol, and cytoskeleton. Each of these structures requires different and specialized experimental tools to perform a useful analysis (Chen, 2014). Moreover, the mechanical properties of cells are also heavily influenced by their ability to deform (a.k.a. their deformability) when they are subjected to an external force such as a shear flow or an acoustic force. Cell deformability plays a vital role in many assays and procedures, including cell separation, disease diagnostics, drug testing, and immunity analysis. Because of this, the study of the mechanical properties of biological cells is of great interest to both academia and industry, particularly for cancer progression analysis (Smelser et al., 2015; Kashani and Packirisamy, 2020). The migration of cancer cells has been shown to be associated with alterations in the cytoskeletal architecture of cells and, consequently, their stiffness, which is one of their mechanical properties. These changes enable cancer cells to migrate and attack distant organs (Kashani and Packirisamy, 2020). Recent studies have confirmed that cancer cells have an elasticity that is significantly lower than that of healthy cells (Kwon et al., 2020; Han et al., 2020). Based on these findings, we aim to design a method to measure the elasticity of a cell and by extension, determine if it is cancerous.

Single-cell mechanics analysis plays an important role when it comes to accurate diagnosis for some diseases like cancer where multi-cell mechanics find it challenging to consider different types of cells, especially the aggressive ones, within the tumor (Kozminsky and Sohn, 2020). However, characterizing the mechanical properties of a single microcapsule or biological cell is difficult due to their fragility and small size. There are several experimental techniques used for measuring

[*] Corresponding author. Department of Mechanical Engineering, University of Wisconsin-Madison, Madison, Wisconsin 53706 -1572, United States.
E-mail address: yli2562@wisc.edu (Y. Li).

[1] These authors contributed equally.

https://doi.org/10.1016/j.mechmat.2023.104631
Received 22 September 2022; Received in revised form 3 March 2023; Accepted 9 March 2023
Available online 11 March 2023

Table 1
Recent machine learning works on biological cells and tissues.

No.	ML models	Application	Input	Main results	Ref.
1	NN	Classification of red blood cells (RBC).	RBC properties (estimated based on quantitative phase imaging), e.g., volume, surface area, etc.	>98% accuracy	Kim et al. (2019)
2	CNNs		Deformed cell images under capillary condition	>80% accuracy	Kihm et al. (2018)
3	SVM		RBC properties (estimated based on quantitative phase imaging with micro-spectrocolorimetry), e.g., area, perimeter, etc.	>95% accuracy	Singh et al. (2020)
4	SVM	Classification of various cell types (cancer cells and blood cells) in blood.	2D morphological features (based on microscopic images)	>80% accuracy	Nissim et al. (2021)
5	06 ML tools: AdaBoost, GB, KNN, RF, and SVM.	Classification of white blood cells	Morphological features (based on microscopic images)	97% average F1-score	Nassar et al. (2019)
6	CNNs	Prediction of collagenous tissue elastic properties	Microscopy images	Classification accuracy of 84%; stress-strain regression errors of 0.021 and 0.031 were achieved	Liang et al. (2017)
7	RF	Prediction of optical property	Diffuse reflectance images in the spatial frequency domain	Errors ~ 0.556% in absorption and 0.126% in reduced scattering.	Panigrahi and Gioux (2018)
8	CNNs	Prediction of the tissue deformation (strain field)	Radio frequency	Compute the strain field robustly and accurately	Wu et al. (2018)
9	DT, ERTs and RF	Prediction of the real-time breast tissue deformation based on finite element (FE) simulations (ground truth).	Magnetic resonance images	Lowest errors correspond with ERTs; 0.4% of the mean displacement and 0.43% of the standard deviation.	Martinez-Martinez et al. (2017)
10	DCNN-LSTM	Prediction of membrane elasticity and membrane viscosity	Sequence of capsule deformed snapshots from dynamics simulations	The mean absolute percentage error of the capillary number and membrane viscosity are 3.65% and 3.42%, respectively.	Lin et al. (2021)

NN = neural networks; CNNs = convolutional neural networks; SVM = support vector machine; GB = gradient boosting; KNN = K-nearest neighbors; RF = random forest; DF = decision tree; ERTs = extremely randomized trees; DCNN-LSTM = deep convolutional neural network - long short-term memory.

single-cell deformability, including atomic force microscope (AFM), micropipette aspiration, parallel-plate technique, magnetic twisting cytometry, magnetic tweezers, optical stretcher, optical tweezers, acoustic methods, and particle-tracking micro-rheology (Unal et al., 2014). Among them, AFM is one of the most popular tools (Haase and Pelling, 2015). Each of these methods is limited by a slow measurement speed, which does not allow for high throughput characterization of biological samples (Darling and Di Carlo, 2015). One method that addresses this shortcoming is the microfluidics method, which has a high-throughput rate in laboratory testing. Due to its ultra-high throughput rate (10^3–10^4 cells/s), microfluidics is commonly used to assess the mechanical properties of thousands of individual cells very quickly, especially the properties related to individual subcellular components. A common use of microfluidics is assessing the deformation of blood cells by putting them under a constricted channel. Through this assay, one could classify all the major blood cell types and pathological changes due to certain disease conditions present in a human blood sample (Bento et al., 2018; Toepfner et al., 2018). Despite the high throughput rate afforded by microfluidics, data analysis methods have historically struggled to keep pace with the large amount of raw data generated. Time consuming image processing and analysis methods limit the effective application of microfluidics (Lin et al., 2021). Fortunately, machine learning (ML) can help us overcome this bottleneck. ML algorithms allow for greatly enhanced image processing speeds. Combining ML image processing algorithms with the high throughput data of microfluidics allows for a versatile tool that can be used for real-time applications (Sarker, 2021). Some of current ML applications in biological field are listed in Table 1.

According to Table 1, even though researchers have put more effort into developing ML tools for the prediction of cell and tissue mechanics, there are only a few ML-based prediction models developed to estimate the mechanical properties of an individual cell. The most recent ML-based work to predict a single capsule's biomechanics was presented by Lin et al. (2021) using a combination of a deep convolutional neural network (DCNN) and a long short-term memory (LSTM), creating a so-called *DCNN-LSTM* network (Lin et al., 2021). This DCNN-LSTM framework predicted the membrane elasticity and viscosity of microcapsules from a series of their dynamic deformation images, taken while they were flowing in a branched microchannel. By doing so, the authors achieved mean absolute percentage errors as low as 3.65% and 3.42% for the predicted capillary number and membrane viscosity, respectively. Even though this work could characterize the viscoelastic properties accurately, we observed that the fluid channel's geometry used in this study is unable to represent standardized laboratory tests in reality, which is mostly focused on the microfluidic approach. Moreover, instead of using sequential images like their approach, we aim to take only the data of the deformed capsule at the end of the constricted channel to characterize the single capsule's mechanical properties. This allows us to follow the experimental setup of microfluidic methods that usually capture the cell's deformation in a region of interest (Urbanska et al., 2020).

Previously, Marta Urbanska et al. (2020) authored a study where microfluidics was used to measure the deformability of individual cells. This inspired us to develop a practical ML-based computational framework for predicting the mechanical properties of a single cell, trained on a dataset we developed. In the present study, we use our recently developed package for fluid-structure interaction (FSI) simulations, so-called OpenFSI (Ye et al., 2020), to generate various coarse-grained microcapsule geometries with different membrane deformability. We then use the FSI simulation-generated dataset as the input for a convolutional neural network (CNN) model with the desired output being the membrane stiffness of the capsules. A CNN can be used as a digital image processing system with the ability to resolve and evaluate the details of an input image. They are commonly used to recognize and classify specific objects or humanoid faces within pictures. Recently, CNN models have become powerful enough to take advantage of the

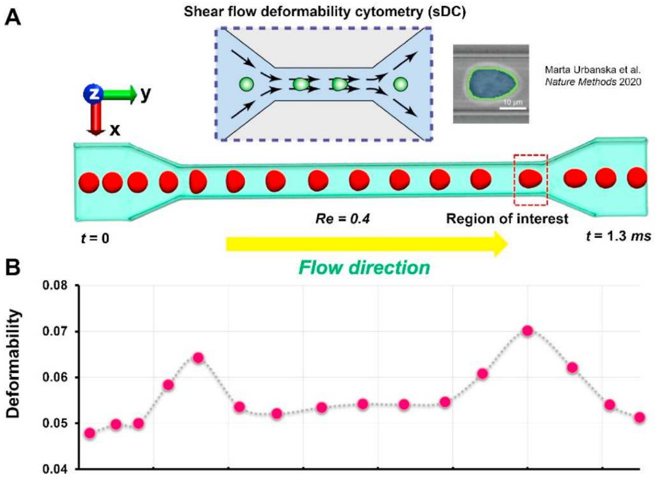

Fig. 1. A model of a deformed capsule in a shear flow deformability cytometry. (A) Time-series snapshots of a capsule's geometry in a shear flow microfluidic channel from t = 0 to t = 1.3 ms. The Reynold number (Re) is set at 0.4 to reproduce the experimental flow condition (Urbanska et al., 2020). The top-view snapshot of the capsule at the end of the channel, so-called region of interest (ROI), shows a bullet-like shape of deformation compared to the experimental image in the inset (Urbanska et al., 2020). **(B)** The corresponding deformability of the capsule along the channel is calculated as $D = 1 - \frac{2\sqrt{\pi\,Area}}{Perimeter}$, showing that the capsule reaches its maximum deformation at the ROI. The 2D snapshot and 3D geometry of the deformed capsule at the ROI are then used to predict the membrane stiffness of the capsule.

large pre-existing dataset of biological cell images obtained from microscopic tools (Allier et al., 2022; Dietler et al., 2020; Oei et al., 2019). Due to their exceptional image processing ability, CNN algorithms have been used in various applications in biological systems including phenotyping single cells directly from microscopy images (Berryman et al., 2020) or assessing blood cell lesions at a level that surpasses human performance (Doan et al., 2020). Specially, U-Net is a powerful CNN-based framework that was recently developed for cell counting, detection, and morphometry (Falk et al., 2019). This study takes advantage of the image processing power of CNNs to measure the stiffness of cell membranes. We show that these CNN-based models can predict the mechanical properties of the single capsule accurately based on its two-dimensional (2D) deformed images and has improved accuracy when its three-dimensional (3D) deformed geometry is available.

In this study, we build an efficient physics-constrained ML model for the sake of performance enhancement. Physics-informed ML models have been widely used to overcome the low data availability of some biological and engineering systems that makes most state-of-the-art ML techniques ineffective. These models add the knowledge of physical laws in the form of partial differential equations and constrain a given dataset in the training process (Raissi et al., 2019). This technique has been used for a wide variety of applications, including physics-informed prediction of the pressure and saturation plumes of multiphase flow (Yan et al., 2022), simulation and synthesis of cyclic adsorption processes (Subraveti et al., 2022), and modeling cyclic voltammetry with electrochemically consistent boundary conditions (Chen et al., 2022). Here, instead of using governing equations during the training process, we take into account physical quantities including total surface area and total volume of the capsule with an assumption that these values do not change after the capsule's deformation. These physical constraints are incorporated into the loss function to enhance the convergence of ML models. We demonstrate that embedding this prior knowledge into a neural network can facilitate the learning process to get accurate results with relatively small datasets. We expect our physics-constrained CNN framework will be used as a novel computational approach to support real-time biomechanical measurement applications in the future.

Table 2
Coarse-grained parameters for spherical microcapsule model with their corresponding physical values.

Parameters	Simulation (LJ unit)	Physical
Energy scale (k_BT)	6.57×10^{-6}	4.14×10^{-21} N m
Viscosity of fluid (h)	0.333	0.0057 Pa s
Channel width × height	20 × 20	20 × 20 μm
Mean flow velocity	0.0068	0.1 m s^{-1}
Applied stress	0.0796	~1 kPa
Reynold number	0.4	0.4
Cell radius	4.0–9.0	4.0–9.0 × 10^{-6} m
Cell shear modulus (μ_r)	0.0016–0.0635	1.0–40 × 10^{-6} N m^{-1}
Cell area constant (k_a)	0.0012–0.0476	0.75–29.97 × 10^{-6} N m^{-1}
Cell local area constant (k_d)	0.0582–2.3280	0.36–14.6 × 10^{-4} N m^{-1}
Cell volume constant (k_v)	0.0627–2.5094	39.5–1580 N m^{-2}
Cell bending constant (k_b)	0.0001–0.0050	0.79–31.75 × 10^{-19} N m

2. Computational model and methods

The present method that predicts the membrane stiffness of a microcapsule from its corresponding dynamic deformation is based on a shear flow deformability cytometry (sDC). In this work, the membrane stiffness in this work corresponds to the membrane shear modulus, which appears for the elastic energy storage connected with the shear deformation of the capsule membrane. The computational framework consists of two parts. The first part is to build a microfluidic platform for a flow-induced capsule deformation, which is detailed in **Section 2.1**. The second part is to develop a CNN-based prediction algorithm. As described in **Section 2.2**, CNN model belongs to supervised learning, and can be used as an operator which establishes the relationship between input and output, which are the deformed geometry of a capsule and its membrane stiffness, respectively.

2.1. Fluid–structure interaction (FSI) computational method

The computational model is shown in Fig. 1A. An initially spherical capsule flows through a microfluidic channel, which is called sDC. In the sDC, the constricted channel has a square cross-section 20 × 20 μm^2 with a length 300 μm that follows the corresponding experimental setup (Urbanska et al., 2020). 3D Cartesian coordinate is used with *y*-axis along the axis of the main channel. To quantify the deformability of the capsule, we calculated the deform value (D) of the capsule as $D = 1 - \frac{2\sqrt{\pi\,Area}}{Perimeter}$ as presented in Ref (Urbanska et al., 2020). for all snapshots (2D top-view images) in Fig. 1A. Here, the deformability is the degree to which applying a force can change the shape of the capsule membrane. As the capsule gets more deformation, the shape ratio ($\frac{2\sqrt{\pi\ \mathrm{Area}}}{\mathrm{Perimeter}}$) of the deformed capsule in the 2D top-view image gets smaller leading to increasing its D values. The length of the constricted region is long enough to allows the capsule to develop its deformation along the channel (Fig. 1B). We define a "region of interest" (ROI) at the end of the channel where the capsule reaches its maximum deformability. The data derived from the deformed capsule (2D top-view snapshot or 3D coordinates of membrane's vertices) in the ROI is then employed by a pre-trained CNN model to predict the membrane stiffness of capsule.

The capsule's membrane is discretized into a point system with triangular networks. The Lagrangian mesh of the membrane is approximately uniform, and the mesh size is about 0.5 μm. The numbers of vertices and elements depend on the size of the capsule. For example, a membrane of a capsule with the radius of 6.5 μm will have 1896 vertices and 3788 elements (Fig. S1). The membrane structure is immersed in an incompressible flow (water). The flow viscosity and mean velocity used in the model are adjusted to match the experimental flow regime (Re = 0.4) (Urbanska et al., 2020) (Table 2 and Table S1). First, the capsule is placed in the center of the left end of a simulation box with the dimension of 60 × 440 × 25 μm^3 (x × y × z). After that, the capsule is driven through the microfluidic channel by applying a body force in the

y-direction to mimic a pressure-driven flow with periodic boundary conditions only applied along this direction.

The FSI computational framework used in this work has been introduced in our previous studies (Ye et al., 2017, 2019a, 2021). Here we just briefly introduce it for the sake of completeness of this study.

2.1.1. Lattice Boltzmann method for fluid flow

The fluid flow in the microfluidic channel is modeled as an incompressible flow and governed by the continuity and the Navier-Stokes (NS) equations as follows:

$$\nabla \bullet \boldsymbol{u} = 0 \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \bullet \nabla \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2 \boldsymbol{u} + \boldsymbol{F} \tag{2}$$

with $\boldsymbol{u}$, ρ, p denoting fluid velocity, density and pressure, respectively. μ and $\boldsymbol{F}$ are the dynamic viscosity of the fluid and the body force, respectively. We use the Lattice Boltzmann method (LBM) to recover the NS equation instead of solving it directly. The details of how LBM can recover NS equation and the limitations can be found in (Chen and Doolen, 1998). LBM is an approach to solve the discrete Boltzmann equation. Based on the correlation between the mesoscopic Boltzmann and continuum NS equations, LBM can efficiently handle fluid dynamics (Chen and Doolen, 1998). Briefly, LBM is built on the Eulerian coordinate system. The basic parameter in LBM is the density distribution function f_i ($\boldsymbol{x}$, t) at position $\boldsymbol{x}$ and time t. The linearized Boltzmann equation is introduced as:

$$(\partial_t + e_{i\alpha}\partial_\alpha)f_i = -\frac{1}{\tau}\left(f_i - f_i^{eq}\right) + F_i \tag{3}$$

where $\boldsymbol{e}_i$ is the lattice velocity in the i-th direction. Here, f_i ($\boldsymbol{x}$, t) is split into two parts: streaming (L.H.S) and collision (R.H.S). The streaming part is discretized as f_i ($\boldsymbol{x} + \mathbf{e}_i$, $t + 1$) - f_i ($\boldsymbol{x}$, t) where f_i ($\boldsymbol{x}$, t) depends on both time and spatial spaces. The collision part is related to the relaxation process at state ($\boldsymbol{x}$, t) towards the equilibrium state of the particle f_i^{eq} through collision behavior. We choose the most popular scheme for the collision model, Bhatnagar–Gross–Krook (BGK) scheme where only relaxation time t is controlled (Qian et al., 1992). Last, F_i represents the external forcing term. Here, we use D_3Q_{19} model, meaning that each point has 19 lattice velocities with different directions in the three-dimensional Eulerian system (Qian et al., 1992). The lattice velocities can be written as:

$$[\mathbf{e_0}, \mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, \mathbf{e_4}, \mathbf{e_5}, \mathbf{e_6}, \mathbf{e_7}, \mathbf{e_8}, \mathbf{e_9}, \mathbf{e_{10}}, \mathbf{e_{11}}, \mathbf{e_{12}}, \mathbf{e_{13}}, \mathbf{e_{14}}, \mathbf{e_{15}}, \mathbf{e_{16}}, \mathbf{e_{17}}, \mathbf{e_{18}}] = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

The equilibrium distribution function $f_i^{eq}(\boldsymbol{x}, t)$ can be estimated using Maxwell distribution as:

$$f_i^{eq}(\boldsymbol{x}, t) = \omega_i \rho \left[1 + \frac{\mathbf{e}_i \bullet \boldsymbol{u}}{c_s^2} + \frac{(\mathbf{e}_i \bullet \boldsymbol{u})^2}{2c_s^4} - \frac{(\boldsymbol{u})^2}{2c_s^2}\right] \tag{4}$$

where weighting coefficients $\omega_i = 1/3$ ($i = 0$), $\omega_i = 1/18$ ($i = 1$–6), $\omega_i = 1/36$ ($i = 7$–18); sound speed $c_s = \Delta x/(\sqrt{3}\ \Delta t)$ with Δx and Δt representing spatial and temporal discretization sizes, respectively.

The dynamic viscosity in equation (2) can be expressed as:

$$\mu = \rho\left(\tau - \frac{1}{2}\right)c_s^2 \Delta t \tag{5}$$

The external forcing term can be discretized as:

$$F_i = \left(1 - \frac{1}{2\tau}\right)\omega_i \left[\frac{\mathbf{e}_i - \boldsymbol{u}}{c_s^2} + \frac{(\mathbf{e}_i \bullet u)}{c_s^4}\mathbf{e}_i\right] \bullet \boldsymbol{F} \tag{6}$$

After each time step, we collect the f_i in the whole domain and calculate the fluid density and momentum using the relations:

$$\rho = \sum_i f_i, \rho \boldsymbol{u} = \sum_i f_i \mathbf{e}_i + \frac{1}{2}\boldsymbol{F}\Delta t \tag{7}$$

2.1.2. Coarse-grained model for capsule

According to some restrictions to the bio-membrane modeling (Tan et al., 2008), there are some assumptions that we imposed to our capsule model:

- o The capsule is spherical.
- o The capsule membrane is incompressible, homogeneous, isotropic, and elastic.
- o The total surface area and total volume of the capsule are kept constant.

The capsule is modeled with coarse-grained method, where the capsule's membrane is considered as point systems connected with specific triangular meshes (Fedosov et al., 2010). The capsule is represented by a 3D liquid-filled membrane immersed in the fluid. A spherical shape is chosen to represent the capsule with its radius varying from 4.0 to 9.0 μm. The capsule's membrane is discretized into a point system with the average distance between points of 0.5 μm. The mechanical properties of the membrane are implemented by applying potential functions on the triangular network, including in-plane and out-of-plane potentials.

A harmonic bond potential U_{bond} is introduced to account for the in-plane interactions between the points within the capsule's membrane:

$$U_{bond} = K(r - r_0)^2 \tag{8}$$

where K is the bonded force constant, and r_0 is the equilibrium bond length.

In addition to the in-plane interaction, the out-of-plane bending of the membrane is applied with a bending potential function:

$$U_{bending} = \sum_{k \in 1\ldots N_s} k_b\left[1 - cos(\theta_k - \theta_0)\right] \tag{9}$$

in which k_b represents the bending stiffness of the membrane, θ_k appears for the dihedral angle between two adjacent triangular elements andθ_0 is the corresponding initial value. N_s stands for the total number of dihedral angles.

We also apply an area conservation constraint for the capsule's membrane as:

$$U_{area} = \sum_{k \in 1\ldots N_t} \frac{k_d(A_k - A_{k0})^2}{2A_{k0}} + \frac{k_a(A_t - A_{t0})^2}{2A_t} \tag{10}$$

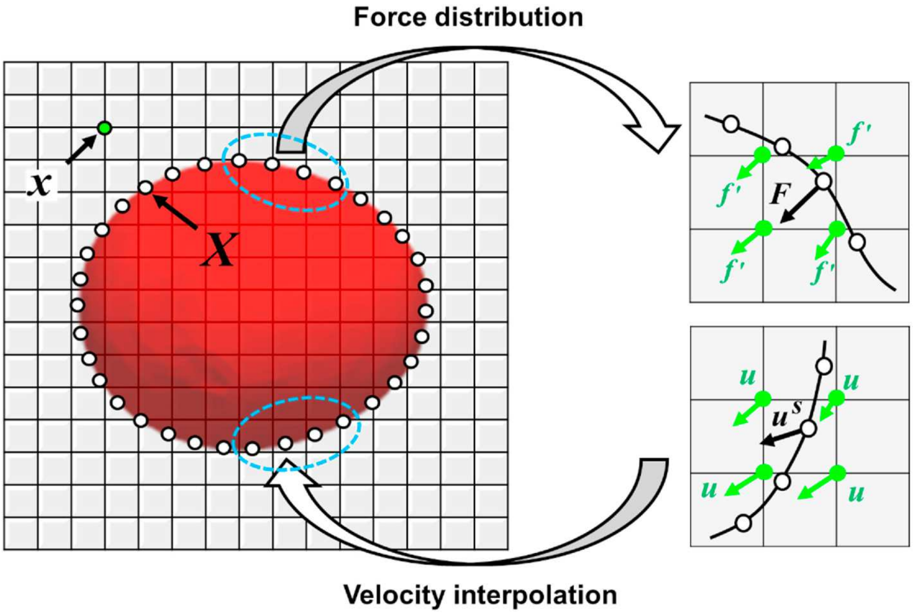

Fig. 2. Schematic of the IBM. Solid squares (grey) represent the Eulerian – surrounding fluid nodes (***x***), and solid circles denote vertices of the Lagrangian structure – capsule membrane nodes (***X***).

The area conservation includes local and global area conservations. The first term represents the local area constraint in which A_k is the k-th element area, A_{k0} is its initial area, k_d is the spring constant, and N_t is the total number of triangular elements. The second term is the global area constraint with spring constant k_a, total area and its initial value A_t and A_{t0}, respectively.

We also assume the volume of the capsule should be constant due to the presence of cytosol inside the cell. Therefore, we employ a simple harmonic function for the capsule's total volume as:

$$U_{volume} = \frac{k_v (V - V_0)^2}{2V_0} \tag{11}$$

with spring constant k_v. total volume V and its initial value V_0, respectively.

Finally, we combine all the potentials to calculate the force at each vertex of the capsule's membrane as:

$$\boldsymbol{f}_i = -\frac{\partial U([\boldsymbol{x}_i])}{\partial \boldsymbol{x}_i} \tag{12}$$

where U ([$\boldsymbol{x}_i$]) is the combination of potentials at coordinate $\boldsymbol{x}_i$.

2.1.3. Coupling of fluid and coarse-grained models: immersed boundary method

Since the capsule is immersed in a fluid flow, the immersed boundary method (IBM) is used to couple the coarse-grained model with the surrounding fluid flow (Ye et al., 2017, 2019b; Liu et al., 2006; Mittal et al., 2008; Zhang et al., 2004; Huang et al., 2007; Tian et al., 2011). In IBM, the coupling is achieved by interpolating the velocity and force at the interface of coarse-grained structure and fluid meshes (Fig. 2). The Lagrangian (moving freely) and Eulerian (fixed) coordinate systems are used to describe the coarse-grained membrane structure and the fluid flow, respectively. The IBM ensures no-slip boundary condition at the interface of the structure and the fluid. The structure (capsule's membrane) is allowed to move with the same velocity as the surrounding fluid, and the force acquired from the coarse-grained model will be spread to the adjacent Eulerian fluid meshes through interpolation, which is considered as an external force term in LBM. We assign the Eulerian coordinates $\boldsymbol{x}$ and Lagrangian coordinates $\boldsymbol{s}$. The structure's position can be denoted as $\boldsymbol{X}$ ($\boldsymbol{s}$,t). The no-slip boundary condition is satisfied as long as:

$$\frac{\partial \boldsymbol{X}(\boldsymbol{s},t)}{\partial t} = u(\boldsymbol{X}(\boldsymbol{s},t)) \tag{13}$$

which means discretized vertices in the coarse-grained model will move with the same velocity as the nearby fluid meshes. Once the vertices move, we then calculate the structure force density $\boldsymbol{F}$ ($\boldsymbol{s}$, t) through potential functions and exert this force to the adjacent fluid meshes by:

$$\boldsymbol{f}^{fsi}(\boldsymbol{x},t) = \int_{\Omega} \boldsymbol{F}(\boldsymbol{X},t)\delta(\boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X},t))d\Omega \tag{14}$$

in which δ is a smoothed approximation for Dirac delta interpolation function. Here, we use the so-called 4-points stencil as:

$$\delta(x) = \begin{cases} \frac{1}{8}\left(3 - 2|x| + \sqrt{1 + 4|x| - 4x^2}\right), 0 \le |x| \le 1 \\ \frac{1}{8}\left(5 - 2|x| + \sqrt{-7 + 12|x| - 4x^2}\right), 1 \le |x| \le 2 \\ 0, 2 \le |x| \end{cases} \tag{15}$$

This stencil takes 64 fluid nodes into account, which has shown stability and fewer artifacts (Peskin, 2002). The force $\boldsymbol{f}^{\text{fsi}}$ ($\boldsymbol{x}$, t) is then used as a body force in the LBM. The same interpolation approach is used to get the velocities of the structure on the moving boundary by:

$$\boldsymbol{u}(X,t) = \int_{\Omega} \boldsymbol{u}(\boldsymbol{x},t)\delta(\boldsymbol{x} - \boldsymbol{x}(\boldsymbol{X},t))d\Omega \tag{16}$$

The channel (sDC) wall is considered as a stationary immersed boundary in the simulations. Our FSI computational framework has been validated by our previous studies (Ye et al., 2017, 2018, 2019b). All computational models were implemented using a recently developed package by Ye and co-workers, the so-called OpenFSI (Ye et al., 2020). Within this package, the structure dynamics are accounted for by a lattice model implemented by Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) (Plimpton, 1995). The parameters of the potentials and corresponding physical parameters of the capsule membrane are presented in Table 2. The coefficients in the potential functions are chosen based on our previous work on red blood cells suspension in a constricted channel (Ye et al., 2021). Freeze is set for the sDC channel and viscous effect is added in the simulation setup. The

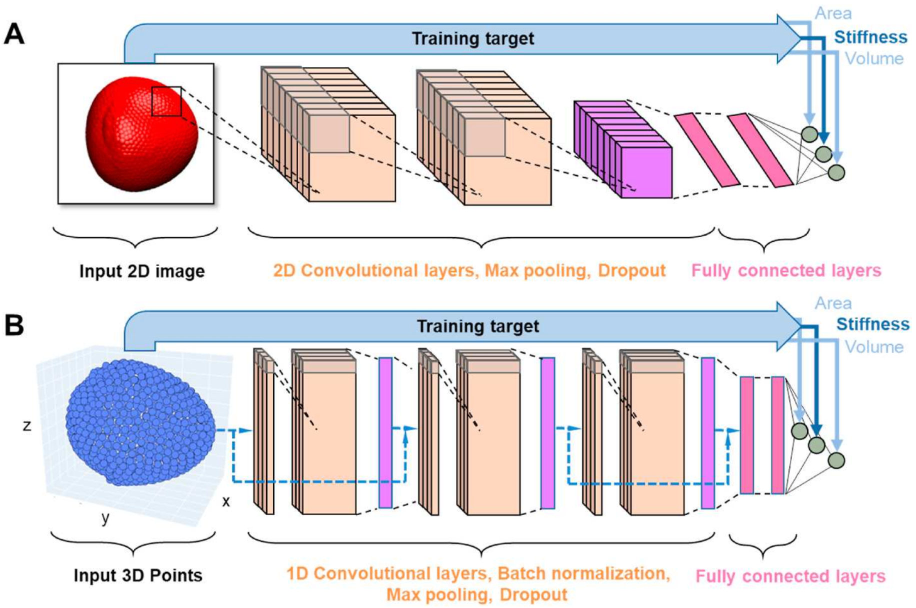

Fig. 3. Architectures of our (A) 2D-CNN and (B) 3D-CNN models. (A) The input of 2D-CNNs model is the top-view snapshot of the deformed capsule at the end of microfluidic channel derived from FSI simulations. The model includes a series of 2D convolutional and max pooling layers that are used for feature extraction. A flattened layer is then used for placing all the extracted features into a single-column feature vector prior to being fully connected in a neural network. The output is the predicted stiffness of the capsule's membrane. In the case of using physical constraints, the output will account for stiffness, total surface area and total volume of the capsule. (B) The input of the 3D-CNN model is the 3D coordinates of the vertices of the capsule's membrane acquired from FSI simulations. The 3D-CNN architecture includes a series of convolutional, batch normalization, activation, global max pooling, and dense layer for feature extractions. The output is the predicted stiffness of the capsule's membrane. Like the 2D model, the outputs of the 3D physical-constrained model will include stiffness, total surface area and total volume of the capsule.

capsule parameters including cell area constant (k_a), cell local area constant (k_d), cell volume constant (k_v) and cell bending constant (k_b) are varied according to the change of membrane stiffness (μ_r). The lattice spacing of fluid field d_x is chosen to be 0.1 μm.

2.2. Machine learning models

2.2.1. Machine learning model's architectures

In this work, CNN models are developed to predict the membrane stiffness of a capsule from its deformed shape. CNN is a neural network of convolutional layers, which have a number of filters. In a typical CNN model, the convolutional layers in connection with pooling layer can repeat to reduce the size of the input image, and then being flattened into a single column prior to be fully connected in a neural network for prediction the output (classification or regression). Filter is a specific kernel that can be used for feature extraction in CNN during the training, and these filters will be learned and optimized during the training process. The filters will slide entirely the image and the output of convolved features is the elementwise scalar product of filter weights and each small region of image (Fig. S2A). Max pooling, normally used in CNN, is the sequential step to reduce the complex which keeps the maximum value in the pooling kernel (Fig. S2B). Going through the convolutional and pooling processes, the input image will be reduced significantly in its dimension, but the object and most important features remain. This is the best merit of using CNN for training perplexing images compared to the other algorithms. The next stage is to flatten the convolved features into a single-column feature vector prior to fully connecting it to a feedforward neural network for predicting the outcomes (Fig. S2C).

There are two CNN models that will be developed in this work. The first CNN model is built to predict the membrane stiffness of a capsule from its deformed 2D image in the ROI, so-called 2D-CNN model. The 2D-CNN model uses a top-view snapshot of the deformed capsule in the ROI acquired from FSI simulations as input for training. As shown in Fig. 3A, our 2D-CNN model includes a series of convolutional, max pooling layers for feature extraction. Eventually, all extracted features are flattened and then fully connected to dense layers. The model is a regression model, and the output is the predicted membrane stiffness. The loss function is to minimize the mean absolute error (MAE) between the predicted stiffness and the ground truth during the training. In the physics-constrained ML models, along with the stiffness, the total surface area and total volume of the capsule are also embedded into the training process based on some restrictions imposed on bio-membrane (Tan et al., 2008). Therefore, in this situation, there will be three outputs instead of one output (Fig. 3).

Table 3
Overview of the used layers in the 2D-CNN model.

Layer	Kernel size [px^2]	Subimage size [px^2]
Input layer	–	400 × 400
Convolutional layer 1.1	10 × 10	391 × 391
Convolutional layer 1.2	4 × 4	388 × 388
Max-pooling layer 1	2 × 2	194 × 194
Convolutional layer 2.1	4 × 4	191 × 191
Convolutional layer 2.2	4 × 4	188 × 188
Max-pooling layer 2	2 × 2	94 × 94
Fully connected layer 3	–	94 × 94
Dense layer 4.1, 4.2, 4.3	–	–
Output layer, regression type	–	1 × 1

In detail, the images of deformed capsules are 2D matrix composed of 0/1. When 1 is used to indicate a capsule pixel, 0 is used to dictate a non-capsule pixel. The size of the 2D matrix is 400 × 400. There will be many 1s in the 2D matrix representing capsules with larger radius, yet there will be many 0s in the 2D matrix representing capsules with smaller radius. In addition, the neighbored 1 and 0 suggests the boundary of capsule. When a filter of a convolutional layer scans through the 2D matrix, the size, boundary, and shape of the capsules are expected to be recognized as key features. To characterize these features, pixels on the boundary of the capsule are more critical than those in the capsule. Fig. S3 demonstrates that key features are related to the boundary of the capsule after the last max pooling layer. Details of the 2D-CNN model's layers are presented in Table 3.

The second CNN model used in this work performs on the 3D geometry of the deformed capsule (3D-CNN). Instead of 2D images, 3D-CNN model uses the 3D coordinates of the vertices of the deformed capsule's membrane derived from FSI simulations as the input (Fig. 3B). One question that arises is whether it is feasible to obtain 3D geometry data of cells in real experiments. While obtaining 2D images of cells is comparatively easier, generating a 3D structure is a more complex task. Nonetheless, it is achievable through experimentation. One promising approach is using a 3D imaging flow cytometer, as demonstrated by Zunming Zhang et al. (2022). This imaging technique captures 3D

Table 4
Overview of the used layers in the 3D-CNN model.

Layer	Output Shape	Layer	Output Shape
InputLayer	3582 × 3	BatchNormalization 8	3582 × 16
Conv1D	3582 × 16	Activation 8	3582 × 16
BatchNormalization	3582 × 16	Conv1D 7	3582 × 32
Activation	3582 × 16	BatchNormalization 9	3582 × 32
Conv1D 1	3582 × 16	Activation 9	3582 × 32
BatchNormalization 1	3582 × 16	GlobalMaxPooling1D 1	32
Activation 1	3582 × 16	Dense 3	32
Conv1D 2	3582 × 32	BatchNormalization 10	32
BatchNormalization 2	3582 × 32	Activation 10	32
Activation 2	3582 × 32	Dense 4	16
GlobalMaxPooling1D	32	BatchNormalization 11	16
Dense	32	Activation 11	16
BatchNormalization 3	32	Dense 5	64
Activation 3	32	Reshape 1	8 × 8
Dense 1	16	Dot 1	3582 × 8
BatchNormalization 4	16	Conv1D 8	3582 × 8
Activation 4	16	BatchNormalization 12	3582 × 8
Dense 2	9	Activation 12	3582 × 8
Reshape	3 × 3	Conv1D 9	3582 × 16
Dot	3582 × 3	BatchNormalization 13	3582 × 16
Conv1D 3	3582 × 8	Activation 13	3582 × 16
BatchNormalization 5	3582 × 8	Conv1D 10	3582 × 32
Activation 5	3582 × 8	BatchNormalization 14	3582 × 32
Conv1D 4	3582 × 8	Activation 14	3582 × 32
BatchNormalization 6	3582 × 8	GlobalMaxPooling1D 2	32
Activation 6	3582 × 8	Dense 6	32
Conv1D 5	3582 × 16	Dense 7	32
BatchNormalization 7	3582 × 16	Dense 8	1
Activation 7	3582 × 16		
Conv1D 6	3582 × 16		

images of cells at a rate of 1000 cells per second. When a cell passes through the laser interrogation area, it is illuminated by a scanning light-sheet with a scanning rate of 200-kHz. A spatial filter placed at the image plane contains a series of spatially positioned pinholes that are aligned with the cell flow direction by a predetermined separation. The photomultiplier tubes (PMTs) detect the emitted light from a specific portion of a cell. Finally, a spatial-temporal transformation is applied to reconstruct the 3D tomographic images.

The 3D-CNN architecture is more complicated compared to that of the 2D-CNN model with a series of convolutional, batch normalization, activation, global max-pooling and dense layers. Similar to 2D-CNN models, the loss function needs to account for the total surface area and total volume of the capsule along with the membrane stiffness in case of using physical constraints. The 3D geometry data is made of a cloud of points on the surface of the deformed capsule. A 2D image can be regarded as the projection of the point cloud on a plane. Therefore, a 3D point cloud keeps more comprehensive information of the deformed capsule than that in a 2D image. This study uses the PointNet model (Qi et al., 2017) to analyze the 3D data. The original purpose of the PointNet model is for object classification. It is found that the max pooling feature of the PointNet makes the model focus on the boundary of the 3D point cloud to better identify their geometries. For our dataset of deformed capsules, the 3D points are already on the surface of the capsule, so it is more straightforward for PointNet model to identify the deformed geometries. To accommodate the PointNet model to the stiffness regression problem, we replace the original classification layer with optimized dense layers before the last output layer for stiffness value. Detail of 3D-CNN model's layers is presented in Table 4.

2.2.2. *Dataset*

The training data for our CNN models contains 736 deformed capsules obtained from the FSI simulations. The capsule's physical parameters spread out in ranges of 4.0–9.0 μm for its radius and 1.0–40.0 μN/m for membrane stiffness. We use the Latin hypercube design (LHD) (Viana, 2016) to effectively sample these capsules from this design space. This sampling technique is a better choice for experimenter to

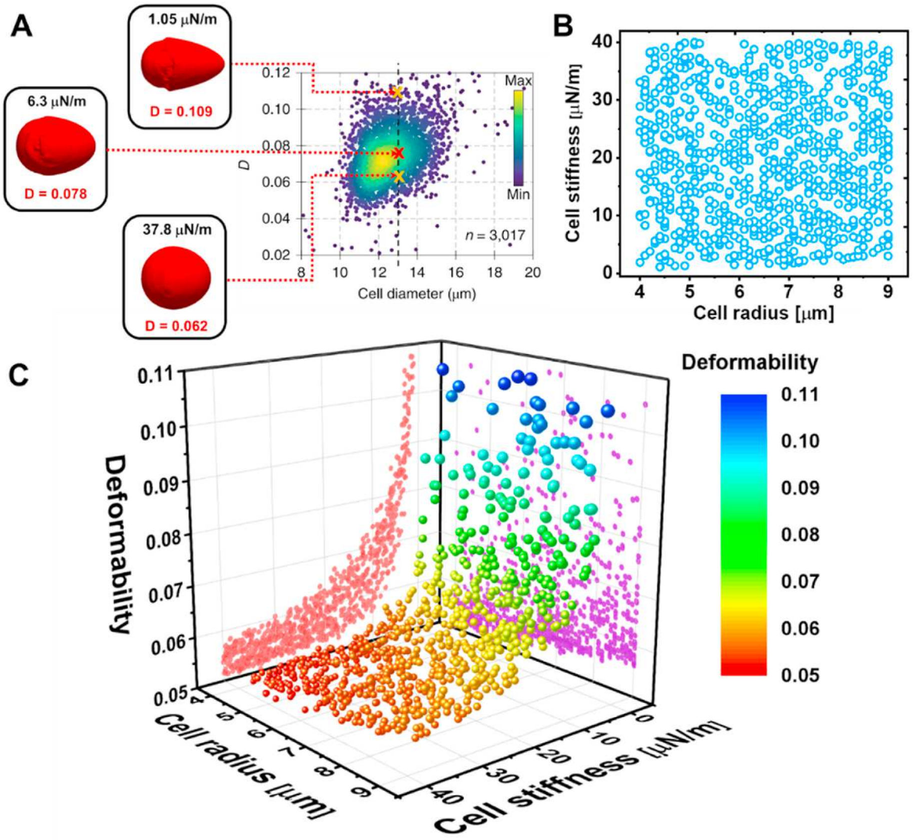

Fig. 4. Capsule's deformability distribution over the dataset used in FSI simulations. (A) Capsules (radius = 6.5 μm) with three different membrane stiffness values show different deformability. These deformed values locate in the range of experimental deform values (scatter plot from (Urbanska et al., 2020)). Snapshots of the deformed capsules in the ROI from the FSI simulations correspond to their membrane stiffness. The capsule with smaller stiffness has higher deformability and vice versa. All snapshots are taken in the ROI and rendered using Paraview (Ahrens et al., 2005; Ayachit, 2015). (B) 736 capsules in the sDC are implemented with FSI simulations. The design of engineering (DOE) points is generated using Latin hypercube sampling method to ensure a well-distributed dataset. Here, two design parameters of a capsule are studied including radius (4.0–9.0 μm) and membrane stiffness (1.0–40.0 μN/m). (C) The 3D plot of distribution of deformability for all 736 capsules obtained from FSI simulations. Here, the deformability is unitless and calculated based on (Urbanska et al., 2020) as $D = 1 - \frac{2\sqrt{\pi\,Area}}{Perimeter}$. The color of each point represents the level of deformability of the deformed cell. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

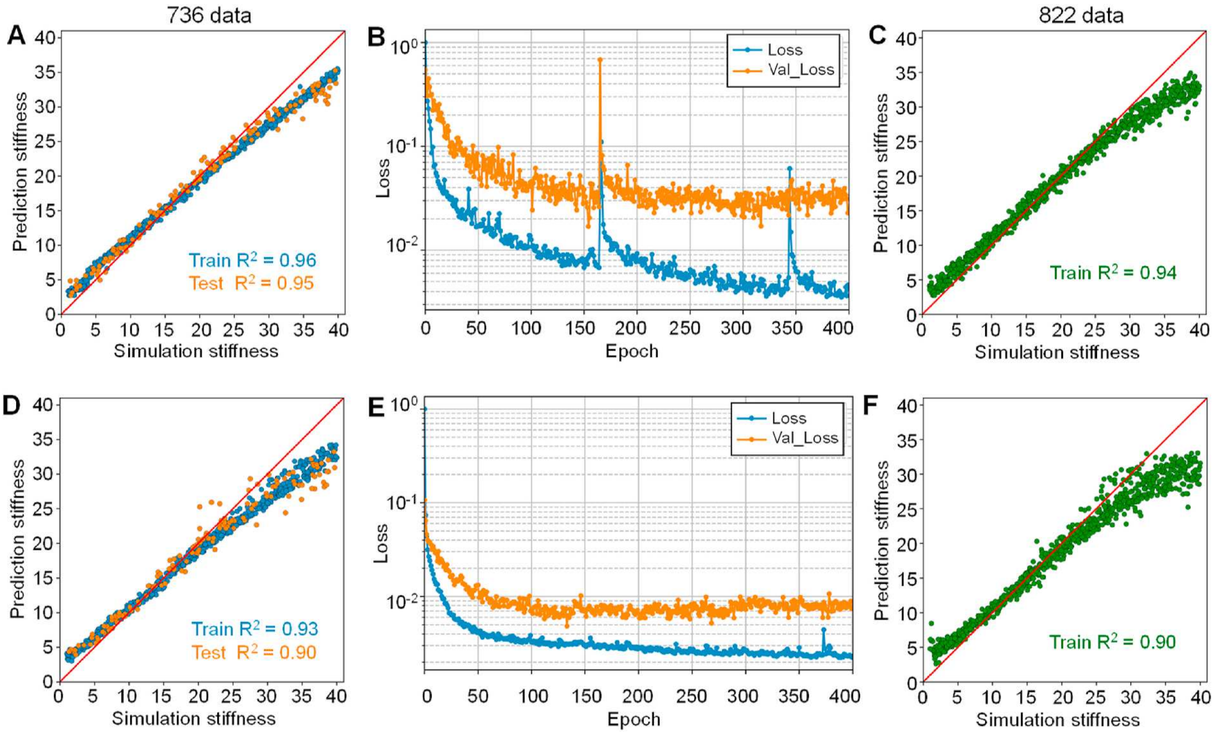

Fig. 5. Ensemble performance results of 2D-CNN models. Comparison of prediction performance between non-constrained (A, B, C) and physical-constrained 2D-CNN models (D, E, F). (A) The CNN model without any constraints performs on training and test datasets with the R^2 of 0.96 and 0.95, respectively. Dataset contains 736 points (train:test = 8:2). (B) The loss (mean absolute error) history of the training and validation verse epochs of non-constrained model. (C) The pretrained model in (A) is validated on an external dataset containing 822 deformed capsules with the R^2 score of 0.94. (D) The CNNs model with total area and volume constraints (as concurrent model's outputs) demonstrates lower values of train and test R^2 scores (0.93 and 0.90) compared to the non-constrained model. (E) The loss history of the training and validation of constrained model in (D) shows more stable than that of non-constrained model. (F) The pre-trained physics-constrained model is validated on the external dataset of 822 data points with the R^2 score of 0.90.

access the impact of the process variables on the output. Rather than building a complete random matrix, LHD tries to subdivide the sample space into smaller cells and choose only one element out of each sub cell. By doing so, a more "uniform spreading" of the random sample points can be obtained. After that, 736 capsule models are generated using our in-house MATLAB code, and then imported into OpenFSI (Ye et al., 2020) to perform FSI simulations. Subsequently, we collected 736 top-view snapshots as well as the 3D geometries of deformed capsules at the ROI for our ML's training and testing. All snapshots are rendered using Paraview (Ahrens et al., 2005; Ayachit, 2015) which performs on the FSI trajectory dump files (LAMMPS trajectory). After establishing the ML models, we followed the same procedure to generate another 822 capsules that are different from the training set for external validation. Thus, we need to run 1558 FSI simulations in total.

3. Results and discussion

3.1. Fluid–structure interaction simulations

The implementation of the FSI model is demonstrated in Fig. 1. The spherical capsule gets deformed when it enters into the constricted channel. The capsule increases its deformation until it reaches the end of the channel (ROI) and then relaxes to its original spherical shape afterward (Fig. 1A). As shown in Fig. 1B, the capsule's deformation reaches its maximum value at the ROI and then decreases when the capsule leaves the channel. The deformed capsule at the ROI is in a bullet-like shape as being observed in the previous experiments (Urbanska et al., 2020), indicating our FSI model successfully reproduced the sDC experiments.

The main goal of this study is to develop an ML-based platform to predict the membrane stiffness of a capsule based on its deformed geometry. To acquire a reliable model, it is important to have a sufficient data source. First, we tested our FSI model with three different capsules having the same radius of 6.5 μm but different membrane stiffness (1.05, 6.3 and 37.8 μN/m). The deformed shapes of the capsules at the ROI were acquired and compared to the previous experimental work (Urbanska et al., 2020) in Fig. 4A. The result confirms that the calculated deform values from the simulation reasonably located in the experimental range of capsule's deformability. After that, 736 data points of capsules with different sizes and stiffness were selected using LHD sampling method over the design space (Fig. 4B). After implementing the fluid dynamics simulations for all the capsules in the sDC, we plotted a 3D distribution of capsule deformability depending on their sizes and stiffness (Fig. 4C). We notice that capsules with smaller membrane stiffness underwent stronger deformation and vice versa (red-point projection plane), while the size effect on the deformation of a capsule seems to be less obvious than that of the stiffness (purple-point projection plane).

3.2. Prediction performance of 2D-CNN models

The ensemble method of three CNN models is used for training to achieve a better prediction performance compared to individual CNN models (Figs. S4–S5) (Džeroski et al., 2009). By taking average of the predictions from multiple models, this ensemble of models reduces the variance and improves the model performance. The number of models to be trained and combined is a hyperparameter which in our cases is optimized as three. Our dataset containing 736 deformed capsule's information is split into the train:test dataset with the ratio of 8:2. Additionally, we aim to compare the prediction performance between non-physical and physical constraint-based ML models. The prediction performance of the non-constraint 2D model is shown in Fig. 5A with the

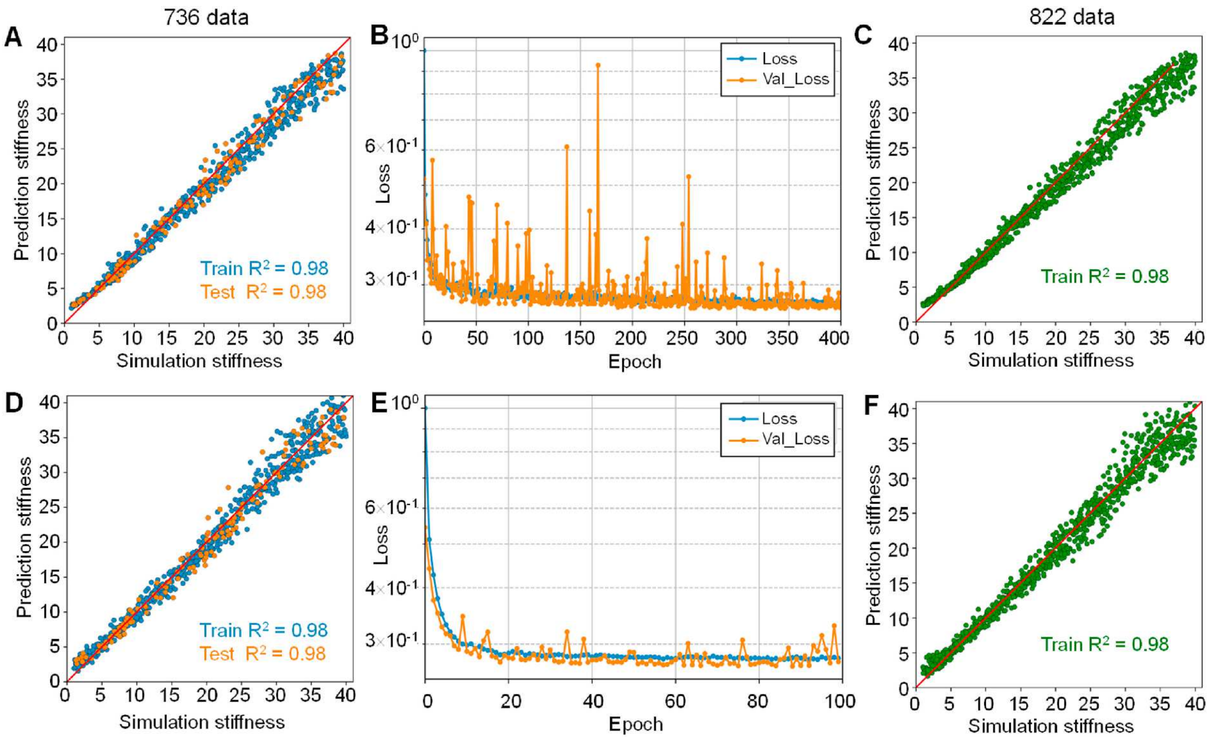

Fig. 6. Ensemble performance results of 3D-CNN models. Comparison of prediction performance between non-constrained (A, B, C) and physical-constrained 3D CNN models (D, E, F). (A) Prediction performance of CNN model without any constraints with R^2 train of 0.98 and R^2 test of 0.98 on the dataset of 736 points (train: test = 8:2). (B) The loss (mean absolute error) history of the training and validation verse epochs of the model in (A). (C) The pretrained model in (A) was validated with external dataset of 822 data points with the R^2 score of 0.98. (D) The model with total area and volume constraints also demonstrates excellent prediction performance with high values of train and test R^2 scores (0.98 and 0.98, respectively) compared to the non-constrained model in (A). (E) The loss history of the training and validation of physical-constrained model in (D) demonstrates better convergence when compared to the non-constrained model. (F) The pre-trained constrained model is validated with the external dataset of 822 data points with the R^2 score of 0.98.

average R^2 scores of 0.96 and 0.95 for the training and testing, respectively. This result indicates that our 2D-CNN model can effectively predict the stiffness of the capsule's membrane solely based on their top-view 2D images. To validate the model, we use an external dataset containing 822 capsule images. This dataset was also acquired from running another 822 FSI simulations. Fig. 5C shows that pretrained model can make an excellent prediction for the stiffness of the capsules in the external dataset with the R^2 score of 0.94. We then examine the predicting performance of the physic-constrained model. As shown in Fig. 5D, even though the constrained model's prediction does not improve compared with the non-constrained one, using the constraints can enhance the convergence of the model in both training and validation. It is indicated in the loss history plot of constrained model (Fig. 5E) when compared with the non-constrained model (Fig. 5B). The constrained model also has a good performance on the external dataset (R^2 score = 0.90) (Fig. 5F). However, both 2D-CNN models still overestimate the stiffness when the true membrane stiffness is low (<10 μN/m) and underestimate the stiffness when the ground truth is high (>25 μN/m) in both training and validation.

3.3. Prediction performance of 3D-CNN models

Due to the underperforming of 2D-CNN models in some ranges of membrane stiffness, we built another type of CNN model that is expected to demonstrate better prediction accuracy. We exploit the geometry of the capsule's membranes as the input for the model, so-called 3D-CNN model. Therefore, the model's input is not a projected image of the deformed shape, but a collection of 3D coordinates of all vertices of the membrane. We hypothesize this 3D information can capture the deformability of the capsule more efficiently than its 2D images. After using ensemble method (Figs. S6–S7), we compared the performance of non-constrained (Fig. 6A) and physical-constrained models (Fig. 6D). We notice that 3D models' average R^2 scores of both training and testing performances are higher than that of 2D-CNN models, indicating the 3D-CNN models are more powerful in predicting the membrane stiffness of these capsules. Additionally, the physical-constrained 3D model outperforms the non-constrained one in terms of the convergence according to their loss history plots (Fig. 6B, **D**). The model with constraints can converge after 100 epochs when compared to the non-constraint model with a significantly longer training process (400 epochs). Eventually, we also validated these 3D models using the external dataset (822 data). These models also present more accurate prediction than the 2D approach with higher R^2 score between the predicted and the ground-truth values (Fig. 6C, **F**). Notably, the 3D-CNN models show that they do not under or over-estimate the membrane stiffness in the high or low true-stiffness domain that previously appeared in the 2D-CNN models. These results demonstrate the merit of using 3D-CNN approach. It is

Table 5
Prediction performance of our CNN models for 736 training and 822 validation capsules.

	2D CNN	2D CNN with constraints	3D CNN	3D CNN with constraints
R^2 training	0.96	0.93	0.98	0.98
R^2 validation	0.94	0.90	0.98	0.98
RMSE training [μN/m]	2.40	3.20	1.60	1.60
RMSE validation [μN/m]	2.66	3.48	1.70	1.72
Max error validation [μN/m]	9.61	12.73	6.62	6.76

more effective for predicting the membrane stiffness and able to overcome the limitations of the 2D-CNN models. However, this 3D approach still faces high uncertainty when the true stiffness of the membrane is elevated (over 30 μN/m).

We summarized the performance of all models in this work in Table 5. Our CNN-based models can perform accurate predictions with the maximum validation R^2 value of 0.98. Additionally, to validate that our proposed method is convincing, we compared our model performance to a baseline model where we derived the membrane stiffness based on the capsule size and deformability using surface fitting (Fig. S8). The validation R^2 value obtained from this simple method is 0.82. Compared to this baseline model, we conclude that our computational approach is powerful to predict the single-cell mechanics by taking advantage of CNN models.

4. Conclusions

In this work, we have presented a computational approach, by integrating high-fidelity FSI simulations and CNN algorithms, for accurate prediction the flowing microcapsule's membrane stiffness. We used our OpenFSI package to generate the training and validating datasets for our ML models. We have demonstrated that the ML models can accurately predict the membrane stiffness in either presence or absence of physical constraints based on either 2D images or 3D coordinates of the deformed membrane. Interestingly, we have shown that our ML models could predict well only based on the static data of deformed capsule at the ROI instead of sequential data points. The present computational framework can be applied to realistic applications of high-throughput single cell characterization since it can reproduce the sDC experimental results. Furthermore, our work also highlighted the importance of adding physical constraints to the ML models for the convergence improvement as well as the utilization of 3D geometry in the CNN model to overcome the underperformance of the 2D models.

We have noticed that our study still needs further work in the near future. First, it needs to improve the underperformance of 2D models at some ranges of stiffness as well as reduce the high uncertainty of 3D models' prediction when the membrane stiffness is high to make the model more robust and reliable. Second, the training dataset has only been derived from FSI simulations, and we have not validated on the experimental images of the cells. Therefore, the future collaboration with some experimentalists those have worked on the sDC/microfluidic methods will be great to validate the capability of ML framework in real-time high-throughput mechanical characterization. Finally, we aim to put more efforts on other characteristics of the cell along with its stiffness for further medical or biological applications, such as monitoring cancer metastasis, vascular disorders, and understanding cell health.

Author contributions

Danh Nguyen and Lei Tao: Conceptualization, Methodology, Software, Data curation, Investigation, Visualization, Validation, Writing-Original draft preparation, Writing-Reviewing and Editing. Huilin Ye: Methodology, Software, Writing-Reviewing and Editing. Ying Li: Supervision, Writing-Reviewing and Editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

Y.L. gratefully acknowledges financial support from the U.S. National Science Foundation (OAC- 1755779, CMMI-1934829 and CAREER Award CMMI-2046751) and 3 M's Non-Tenured Faculty Award. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. National Science Foundation. This research also benefited in part from the computational resources and staff contributions provided by the Booth Engineering Center for Advanced Technology (BECAT) at the University of Connecticut. The authors also acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin (Frontera project and National Science Foundation Award 1818253) and National Renewable Energy Laboratory (Eagle Computing System) for providing HPC resources that have contributed to the research results reported within this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.mechmat.2023.104631.

References

Ahrens, J., Geveci, B., Law, C., 2005. 36 - ParaView: an end-user tool for large-data visualization. In: Hansen, C.D., Johnson, C.R. (Eds.), Visualization Handbook. Butterworth-Heinemann, Burlington, pp. 717–731.

Allier, C., et al., 2022. CNN-based cell analysis: from image to quantitative representation. Frontiers in Physics 9.

Ayachit, U., 2015. The ParaView Guide: A Parallel Visualization Application. Kitware, Inc.

Bento, D., et al., 2018. Deformation of red blood cells, air bubbles, and droplets in microfluidic devices: flow visualizations and measurements. Micromachines 9 (4).

Berryman, S., et al., 2020. Image-based phenotyping of disaggregated cells using deep learning. Commun Biol 3 (1), 674.

Chen, J., 2014. Nanobiomechanics of living cells: a review. Interface Focus 4 (2), 20130055.

Chen, S., Doolen, G.D., 1998. Lattice Boltzmann method for fluid flows. Annu. Rev. Fluid Mech. 30 (1), 329–364.

Chen, H., Katelhon, E., Compton, R.G., 2022. Predicting voltammetry using physics-informed neural networks. J. Phys. Chem. Lett. 13 (2), 536–543.

Darling, E.M., Di Carlo, D., 2015. High-throughput assessment of cellular mechanical properties. Annu. Rev. Biomed. Eng. 17, 35–62.

Dietler, N., et al., 2020. A convolutional neural network segments yeast microscopy images with high accuracy. Nat. Commun. 11 (1), 5723.

Doan, M., et al., 2020. Objective assessment of stored blood quality by deep learning. Proc. Natl. Acad. Sci. U. S. A. 117 (35), 21381–21390.

Džeroski, S., Panov, P., Ženko, B., 2009. Machine learning, ensemble methods in. In: Meyers, R.A. (Ed.), Encyclopedia of Complexity and Systems Science. Springer New York, New York, NY, pp. 5317–5325.

Falk, T., et al., 2019. U-Net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16 (1), 67–70.

Fedosov, D.A., Caswell, B., Karniadakis, G.E., 2010. Systematic coarse-graining of spectrin-level red blood cell models. Comput. Methods Appl. Mech. Eng. 199 (29–32).

Haase, K., Pelling, A.E., 2015. Investigating cell mechanics with atomic force microscopy. J. R. Soc. Interface 12 (104), 20140970.

Han, Y.L., et al., 2020. Cell swelling, softening and invasion in a three-dimensional breast cancer model. Nat. Phys. 16 (1), 101–108.

Hao, Y., et al., 2020. Mechanical properties of single cells: measurement methods and applications. Biotechnol. Adv. 45, 107648.

Huang, W.-X., Shin, S.J., Sung, H.J., 2007. Simulation of flexible filaments in a uniform flow by the immersed boundary method. J. Comput. Phys. 226 (2), 2206–2228.

Kashani, A.S., Packirisamy, M., 2020. Cancer cells optimize elasticity for efficient migration. R. Soc. Open Sci. 7 (10), 200747.

Kihm, A., et al., 2018. Classification of red blood cell shapes in flow using outlier tolerant machine learning. PLoS Comput. Biol. 14 (6), e1006278.

Kim, G., et al., 2019. Learning-based screening of hematologic disorders using quantitative phase imaging of individual red blood cells. Biosens. Bioelectron. 123, 69–76.

Kozminsky, M., Sohn, L.L., 2020. The promise of single-cell mechanophenotyping for clinical applications. Biomicrofluidics 14 (3), 031301.

Kwon, S., et al., 2020. Comparison of cancer cell elasticity by cell type. J. Cancer 11 (18), 5403–5412.

Liang, L., Liu, M., Sun, W., 2017. A deep learning approach to estimate chemically-treated collagenous tissue nonlinear anisotropic stress-strain responses from microscopy images. Acta Biomater. 63, 227–235.

Lin, T., et al., 2021. A neural network-based algorithm for high-throughput characterisation of viscoelastic properties of flowing microcapsules. Soft Matter 17 (15), 4027–4039.

Liu, W.K., et al., 2006. Immersed finite element method and its applications to biological systems. Comput. Methods Appl. Mech. Eng. 195 (13–16), 1722–1749.

Martinez-Martinez, F., et al., 2017. A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time. Comput. Biol. Med. 90, 116–124.

Mittal, R., et al., 2008. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. J. Comput. Phys. 227 (10), 4825–4852.

Nassar, M., et al., 2019. Label-free identification of white blood cells using machine learning. Cytometry 95 (8), 836–842.

Nissim, N., et al., 2021. Real-time stain-free classification of cancer cells and blood cells using interferometric phase microscopy and machine learning. Cytometry 99 (5), 511–523.

Oei, R.W., et al., 2019. Convolutional neural network for cell classification using microscope images of intracellular actin networks. PLoS One 14 (3), e0213626.

Panigrahi, S., Gioux, S., 2018. Machine learning approach for rapid and accurate estimation of optical properties using spatial frequency domain imaging. J. Biomed. Opt. 24 (7), 1–6.

Peskin, C.S., 2002. The immersed boundary method. Acta Numer. 11, 479–517.

Plimpton, S., 1995. Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. 117 (1), 1–19.

Qi, C., et al., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 77–85, 2017.

Qian, Y.H., D'Humières, D., Lallemand, P., 1992. Lattice BGK models for Navier-Stokes equation. Europhys. Lett. 17 (6), 479–484.

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

Rodriguez, M.L., McGarry, P.J., Sniadecki, N.J., 2013. Review on cell mechanics: experimental and modeling approaches. Appl. Mech. Rev. 65 (6).

Sarker, I.H., 2021. Machine learning: algorithms, real-world applications and research directions. SN Comput Sci 2 (3), 160.

Singh, V., Srivastava, V., Mehta, D.S., 2020. Machine Learning-Based Screening of Red Blood Cells Using Quantitative Phase Imaging with Micro-spectrocolorimetry. Optics & Laser Technology, p. 124.

Smelser, A.M., et al., 2015. Mechanical properties of normal versus cancerous breast cells. Biomech. Model. Mechanobiol. 14 (6), 1335–1347.

Subraveti, S.G., et al., 2022. Physics-based neural networks for simulation and synthesis of cyclic adsorption processes. Ind. Eng. Chem. Res. 61 (11), 4095–4113.

Tan, Y., et al., 2008. Mechanical modeling of biological cells in microinjection. IEEE Trans. NanoBioscience 7 (4), 257–266.

Tian, F.B., et al., 2011. An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments. J. Comput. Phys. 230 (19), 7266–7283.

Toepfner, N., et al., 2018. Detection of Human Disease Conditions by Single-Cell Morpho-Rheological Phenotyping of Blood, vol. 7. Elife.

Unal, M., et al., 2014. Micro and nano-scale technologies for cell mechanics. Nanobiomedicine (Rij) 1, 5.

Urbanska, M., et al., 2020. A comparison of microfluidic methods for high-throughput cell deformability measurements. Nat. Methods 17 (6), 587–593.

Viana, F.A.C., 2016. A tutorial on Latin hypercube design of experiments. Qual. Reliab. Eng. Int. 32 (5), 1975–1985.

Wu, S., et al., 2018. Direct reconstruction of ultrasound elastography using an end-to-end deep neural network. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. Springer International Publishing, Cham.

Yan, B., et al., 2022. A physics-constrained deep learning model for simulating multiphase flow in 3D heterogeneous porous media. Fuel 313.

Ye, H., Shen, Z., Li, Y., 2017. Computational modeling of magnetic particle margination within blood flow through LAMMPS. Comput. Mech. 62 (3), 457–476.

Ye, H., Shen, Z., Li, Y., 2018. Shear rate dependent margination of sphere-like, oblate-like and prolate-like micro-particles within blood flow. Soft Matter 14 (36), 7401–7419.

Ye, H., Shen, Z., Li, Y., 2019a. Multiscale modeling of vascular dynamics of micro- and nano-particles. In: Application to Drug Delivery System. Morgan & Claypool Publishers.

Ye, H., Shen, Z., Li, Y., 2019b. Interplay of deformability and adhesion on localization of elastic micro-particles in blood flow. J. Fluid Mech. 861, 55–87.

Ye, H., et al., 2020. OpenFSI: a highly efficient and portable fluid–structure simulation package based on immersed-boundary method. Comput. Phys. Commun. 256.

Ye, H., et al., 2021. Red blood cell hitchhiking enhances the accumulation of nano- and micro-particles in the constriction of a stenosed microvessel. Soft Matter 17 (1), 40–56.

Zhang, L., et al., 2004. Immersed finite element method. Comput. Methods Appl. Mech. Eng. 193 (21–22), 2051–2067.

Zhang, Z., et al., 2022. A high-throughput technique to map cell images to cell positions using a 3D imaging flow cytometer. Proc. Natl. Acad. Sci. U. S. A. 119 (8).