Deep operator network models for predicting post-burn contraction ☆

Selma Husanovic [a], Ginger Egberts [b,c,d], Alexander Heinlein [a,*], Fred Vermolen [b]

[a] Delft Institute of Applied Mathematics (DIAM), Faculty of Electrical Engineering, Mathematics & Computer Science, Delft University of Technology, Mekelweg 4, 2628, CD, Delft, the Netherlands
[b] Research Group Computational Mathematics (CMAT), Department of Mathematics and Statistics, University of Hasselt, Hasselt, Belgium
[c] Department of Plastic, Reconstructive & Hand Surgery, Amsterdam UMC, Amsterdam, the Netherlands
[d] Dutch Burns Foundation, Beverwijk, the Netherlands

ABSTRACT

Background: Burn injuries present a significant global health challenge. Among the most severe long-term consequences are *contractures*, which can lead to functional impairments and disfigurement. Understanding and predicting the evolution of post-burn wounds is essential for developing effective treatment strategies. Traditional mathematical models, while accurate, are often computationally expensive and time-consuming, limiting their practical application. Recent advancements in machine learning, particularly in deep learning, offer promising alternatives for accelerating these predictions.

Methods: This study explores the use of a *deep operator network*, a type of neural operator, as a surrogate model for finite element simulations aimed at predicting post-burn contraction across multiple wound shapes. A deep operator network was trained on three distinct initial wound shapes, with enhancements made to the architecture by incorporating initial wound shape information and applying sine augmentation to enforce boundary conditions.

Findings: The performance of the trained deep operator network was evaluated on a test set including finite element simulations based on convex combinations of the three basic wound shapes. The model achieved an R^2 score of 0.99, indicating strong predictive accuracy and generalization. Moreover, the model provided reliable predictions over an extended period of up to one year, with speedups of up to 128-fold on the Central Processing Unit and 235-fold on the Graphical Processing Unit, compared to the numerical model.

Interpretation: These findings suggest that deep operator networks can effectively serve as a surrogate for traditional finite element methods in simulating post-burn wound evolution, with potential applications in medical treatment planning.

1. Introduction

Burns occur frequently worldwide and are the fifth non-fatal cause of childhood injuries. The number of burn injuries worldwide was nearly 11 million in 2004, and about 180,000 individuals die as a result of severe burn injuries each year (WHO, 2018). Pain, itching, loss of energy, and psychological factors, among others, cause a decrease in the quality of life among patients who survive burn accidents. Extreme pain, infection, hypertrophic scars, and skin contractures, where skin contracts so intensively that patients lose mobility, remain major challenges in burn treatment (Wang et al., 2018). The problem is aggravated by the fact that skin is the largest organ of the human body. Its weight is about 15% of the total body weight. Roughly speaking, skin consists of three layers: the epidermis (top layer), the dermis, and the subcutis. The subcutis is connected to underlying fat, organs, and muscles. If only the epidermis is damaged, then full healing will occur by the proliferation and migration of keratinocytes, which populate the epidermis. If deeper layers, such as the dermis or even the subcutis, are damaged, then the repair mechanism entails the regeneration of the dermis through collagen secretion by fibroblasts and the regeneration of the small blood vessel (capillary) network by revascularization. Serious burn injuries are characterized by damage to multiple skin (dermal) layers and over large areas of the human body. The healing of these large burn injuries often goes along with *dermal contraction* and/or *hypertrophy*. These

---

mechanisms are mainly caused by fibroblasts' (strain-induced) differentiation to myofibroblasts (Desmoulière et al., 1993). While fibroblasts produce collagen type 1, the natural collagen in undamaged (embryonic) skin, myofibroblasts mainly secrete collagen type 3 at a large turnover per unit of time. Furthermore, myofibroblasts exert pulling forces, eventually making the wound area contract. At the same time, the epidermis is closed by the regeneration and migration of keratinocytes. Clinicians refer to the injury as a *wound* during epidermal healing, after which the injury is referred to as a *scar*. Although epidermal closure and dermal contraction influence each other, their exact interplay is still unclear, and therefore, we will not consider this interaction in the current study.

Dermal contraction may reduce the dermal pliability to such an extent that joints lose their mobility, and patients can no longer exercise or perform daily routines. In these cases, the occurred contraction is referred to as a *contracture* (Goel and Shrivastava, 2010). Another side effect that patients with serious burn injuries endure is hypertrophy of scars (Tomasek et al., 2002). This hypertrophy is characterized by a changed dermal topography, which generally changes the appearance of the skin.

While serious burn injuries often resulted in patient death in the past, nowadays, more and more patients survive severe burns. This is why current health care has shifted its focus from survival to the increase of quality of life of the survived patients. Therefore, clinical treatments are directed towards changing cellular behavior such that contraction and deposition of type 3 collagen are mitigated as much as possible. This mitigation is often achieved through methods such as dressings, clothing, skin grafting (skin transplantation by depositing epidermal parts from other body parts on the injured part), and splinting.

Many scientific studies, both clinical and laboratory, have been carried out and are ongoing to improve health care for burn injuries. In order to improve health care, one aims to steer the behavior of involved cells, and therefore, it is crucially important to understand the chain of biophysical mechanisms responsible for the evolution of post-burned skin. This knowledge needs to be validated using laboratory experiments and clinical observations, which represent data in numbers and patterns, and hence, the developed theory needs to be quantified. This quantification proceeds by using mathematical models based on either stochastic (random) processes and partial differential equations (continuum models) or by combinations of the two (Barocas and Tranquillo, 1997; Dallon et al., 1999; Koppenol, 2017; Menon et al., 2017; Stéphanou and Volpert, 2015). Among these, mechanistic models play a crucial role in understanding the underlying biophysical processes of burn injury and healing. To this extent, the current paper involves mathematical models in terms of partial differential equations (PDEs). These equations contain input parameters, such as elasticity and cell division rates, that are unknown beforehand due to patient-to-patient variations. These variations are commonly caused by differences in cell division rates, chemical environment, and mechanical parameters, such as the stiffness of the skin. All these variations are caused by age, gender, pigmentation, genetic composition and life style of the patient. The patient-to-patient variations make the uncertainty in the outcomes of the model large, even in the hypothetical case that the mathematical model contains all the relevant aspects that determine post-burn dermal evolution. For this reason, given the inherent patient-to-patient variations and the complexity of the biological processes, one simulation run cannot be deemed reliable in general. Therefore, the focus shifts to understanding the likelihood of clinical successes under different therapies. This can only be achieved by sampling the input parameters, such as initial wound geometry and model input parameters, from prior probability distributions to estimate posterior probability distributions for the output parameters, such as the extent of wound contraction or strain energy.

One of the bottlenecks in making current simulation tools applicable to clinical practice is the relatively long simulation time required to run one sample. These long simulation times are caused by the non-linear nature of the set of partial differential equations of which the solution is approximated by a moving finite element method combined with time-integration and fixed point problem solvers. To address this issue, neural network-based frameworks are being developed, where a regression-like model links input parameters to the output parameters of interest. Ideally, the resulting model can generate simulation runs at very large speedups compared to classical numerical simulations. The first neural networks to reproduce post-burn contraction finite element simulations in one and two dimensions have been introduced in (Egberts et al., 2022; Egberts et al., 2023). These studies indicate high fidelity and high-speed reproduction of the finite element simulations. However, a limitation of these works is that all simulations were performed on a fixed spatial domain, meaning the wound's size and shape remain constant. This is not very representative of real-world scenarios.

In this study, we propose using *neural operators* to generate a surrogate model that reproduces the results of finite-element simulations, predicting post-burn wound evolution over time. Neural operators are a class of deep learning architectures with the primary application of learning surrogate maps for the solution operators of PDEs. Specifically, we will focus on *deep operator networks (DeepONets)* for learning operators accurately and efficiently from a relatively small dataset (Lu et al., 2021). In particular, we aim to learn the solution operator of the two-dimensional *morphoelastic model*, which is a type of mechanistic model that describes post-burn skin evolution, thus accurately reproducing the finite element predictions. A key aspect of this research is incorporating multiple initial wound shapes. We train the neural operator network on a set of basic wound shapes and evaluate its fidelity using convex combinations of the basic wound shapes as input. In this way, our research aims to extend the works (Egberts et al., 2022; Egberts et al., 2023).

The paper is organized as follows. Section 2 describes the methodology, focusing on numerical simulations and the machine learning models used. Section 3 introduces the datasets and training setup for our neural network. The results are presented in Section 4. Finally, Sections 5 and 6 provide the conclusions and discussion, respectively.

2. Methodology

This sections details the methodology, describing both the numerical simulations and the machine learning models employed.

2.1. Numerical simulations

We first present the mathematical model for post-burn contraction, starting with the general system of equations and then providing a more detailed description of the relevant biological and mechanical components. Furthermore, we give the boundary conditions, initial conditions, and numerical solver and introduce an important measure of contraction. We emphasize that the current model was used and introduced in some of our earlier works (Egberts et al., 2023; Koppenol and Vermolen, 2017) and, therefore, only a short model description is provided in this paper.

2.1.1. Mathematical morphoelastic model

The general morphoelastic model for post-burn contraction was developed by Koppenol (Koppenol et al., 2017), who used the theory of morphoelasticity developed by Hall (Hall, 2008) to incorporate the formation of long-term deformations (contraction) into the dermal layer of the skin.

The model considers four biological constituents and three mechanical components as the primary variables. The biological constituents are the fibroblasts (N), the myofibroblasts (M), a generic signaling molecule (c), and collagen (ρ). The mechanical components are the dermal layer displacement ($\mathbf{u}$), the dermal layer displacement velocity ($\mathbf{v}$), and the effective strain ($\boldsymbol{\varepsilon}$). The following system of partial differential equations is used as a basis for the model:

$$\frac{\mathrm{D}z_i}{\mathrm{D}t} + z_i(\nabla\cdot\mathbf{v}) = -\nabla\cdot\mathbf{J}_i + R_i, \tag{1}$$

$$\rho_t\left(\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} + \mathbf{v}(\nabla\cdot\mathbf{v})\right) = \nabla\cdot\boldsymbol{\sigma} + \mathbf{f}, \tag{2}$$

$$\frac{\mathrm{D}\boldsymbol{\varepsilon}}{\mathrm{D}t} + \boldsymbol{\varepsilon}\,\mathrm{skw}(\nabla\mathbf{v}) - \mathrm{skw}(\nabla\mathbf{v})\boldsymbol{\varepsilon}$$

$$+ (\mathrm{tr}(\boldsymbol{\varepsilon}) - 1)\mathrm{sym}(\nabla\mathbf{v}) = -\mathbf{G}, \tag{3}$$

The operator $\frac{\mathrm{D}}{\mathrm{D}t}$ stands for the *material derivative*:

$$\frac{\mathrm{D}}{\mathrm{D}t} \equiv \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla.$$

Eq. (1) is the conservation equation for the cell density/concentration for each of the four biological constituents. Here, z_i represents the concentration, $\mathbf{J}_i$ is the flux per unit area, and R_i is a reaction term representing the kinetics of constituent i, for $i \in \{N, M, c, \rho\}$.

Eq. (2) is the conservation equation for linear momentum, where ρ_t represents the total mass density of the dermal tissue, σ is the stress tensor, and $\mathbf{f}$ is the total body force working on the dermal layer. Furthermore, $\mathbf{v} = \frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}$. We note that Eq. (2) actually gives rise to multiple equations, one for each component of the velocity vector $\mathbf{v}$. For example, in 2D, this results in two equations.

Finally, Eq. (3) is the evolution equation that describes how the infinitesimal effective strain ($\boldsymbol{\varepsilon}$) changes over time. This equation captures the morphoelasticity of the dermal layer, taking into account permanent deformation (in this case, contraction) and residual stresses. It was formulated by Hall (Hall, 2008) and is based on his extensive theory on the zero stress state and morphoelasticity. The second-order tensor $\mathbf{G}$ is a growth tensor that describes the rate of active change of the effective strain.

2.1.2. The fibroblasts

To simplify notation, we will from now on replace z_i by i. Hence, z_N becomes N, the cell density of the fibroblasts in the dermis. The appropriate flux-term $\mathbf{J}_N$ incorporates both the random movement of fibroblasts through the dermal layer and the directed movement of fibroblasts up the gradient of signaling molecule c, if present. The former is modelled by a cell density-dependent Fickian diffusion, and the latter process is modelled using a simple model for chemotaxis (Hillen and Painter, 2009). Taken together, this gives

$$\mathbf{J}_N = -D_F F\nabla N + \chi_F N\nabla c,$$

where $F = N + M$, D_F is the (myo)fibroblast diffusion parameter, and χ_F is the chemotactic parameter.

Eq. (1) also contains a reaction term R_N describing the kinetics of the fibroblasts. Three processes are considered: proliferation, differentiation into myofibroblasts, and apoptosis. The first is modelled using an adjusted logistic growth model. The presence of a signaling molecule c is assumed to enhance both proliferation and cell differentiation:

$$R_N = r_F\left(1 + \frac{r_F^{\max}c}{a_c^I + c}\right)(1 - \kappa_F F)N^{1+q} - k_F cN - \delta_N N.$$

Here, the parameter r_F is the cell division rate, $r_F^{\max}$ is the maximum factor with which the cell division rate can be enhanced due to the presence of the signaling molecule, and a_c^I is the concentration of the signaling molecule that causes the half-maximum enhancement of the cell division rate. Furthermore, $\kappa_F F$ represents the reduction in the cell division rate due to crowding, q is a fixed constant, k_F is the signaling molecule-dependent cell differentiation rate of fibroblasts into myofibroblasts, and δ_N is the apoptosis rate of fibroblasts.

2.1.3. The myofibroblasts

For the myofibroblasts, the flux term in Eq. (1) is very similar to the one for the fibroblasts:

$$\mathbf{J}_M = -D_F F\nabla M + \chi_F M\nabla c.$$

The reaction term describing the kinetics of myofibroblasts is also very similar. Almost the same adjusted logistic growth model is used, the only difference being the assumption that myofibroblasts solely divide when the generic signaling molecule is present:

$$R_M = r_F\left(\frac{(1 + r_F^{\max})c}{a_c^I + c}\right)(1 - \kappa_F F)M^{1+q} - k_F cM - \delta_M M,$$

where δ_M is the apoptosis rate of myofibroblasts.

2.1.4. The signaling molecules

We assume that the signaling molecules diffuse through the dermis according to linear Fickian diffusion:

$$\mathbf{J}_c = -D_c\nabla c,$$

where D_c is the diffusion coefficient of the generic signaling molecule.

Furthermore, we assume that both fibroblasts and myofibroblasts release and consume the signaling molecules. Additionally, signaling molecules are removed from the dermis through proteolytic breakdown (breakdown of proteins into smaller components). Hence, we obtain the reaction term

$$R_c = \frac{k_c(N + \eta^I M)c}{a_c^{II} + c} - \delta_c g(N, M, c, \rho)c,$$

where k_c is the maximum net secretion rate of the signaling molecule, η^I is the ratio of myofibroblasts to fibroblasts in the maximum net secretion rate of the signaling molecules and the collagen molecules, a_c^{II} is the concentration of the signaling molecule that causes the half-maximum net secretion rate of the signaling molecule, and δ_c is the proteolytic breakdown rate of the signaling molecule.

The function $g(N, M, c, \rho)$ represents the equilibrium concentration of a generic metalloproteinase (MMP). This enzyme is assumed to remove the signaling molecules through a proteolytic breakdown. In this study, we take the following relationship:

$$g(N, M, c, \rho) = \frac{(N + \eta^{II} M)\rho}{1 + a_c^{III} c}.$$

The parameter η^{II} is the ratio of myofibroblasts to fibroblasts in the secretion rate of the MMPs, and the $1/(1 + a_c^{III} c)$ term represents the inhibition of the secretion of the MMPs due to the presence of the signaling molecule.

2.1.5. The collagen molecules

For collagen, we assume no active transport in the dermis. This means that the flux-term in Eq. (1) is zero:

$$\mathbf{J}_\rho = \mathbf{0}.$$

The reaction term includes three processes: collagen molecules are produced by both fibroblasts and myofibroblasts, the secretion rate is enhanced in the presence of the signaling molecule, and there is an MMP-induced proteolytic collagen breakdown analogous to the removal of the signaling molecule:

$$R_\rho = k_\rho\left(1 + \frac{k_\rho^{\max}c}{a_c^{IV} + c}\right)(N + \eta^I M) - \delta_\rho g(N, M, c, \rho)\rho.$$

Here, k_ρ is the collagen molecule secretion rate, $k_\rho^{\max}$ is the maximum factor with which the secretion rate can be enhanced due to the presence of the signaling molecule, a_c^{IV} is the concentration of the signaling molecule that causes the half-maximum enhancement of the secretion rate, and δ_ρ is the degradation rate of the collagen molecules.

2.1.6. The mechanical components

In Eq. (2), a visco-elastic constitutive relation is used for the stress-strain relation in the dermal layer. The visco-elastic relation for the dermal stress is:

$$\boldsymbol{\sigma} = \mu_1 \text{sym}(\nabla\mathbf{v}) + \mu_2[\text{tr}(\text{sym}(\nabla\mathbf{v})\,)\mathbf{I}] + \frac{E\sqrt{\rho}}{1+\nu}\left(\boldsymbol{\varepsilon} + \text{tr}(\boldsymbol{\varepsilon})\frac{\nu}{1-2\nu}\mathbf{I}\right),$$

where μ_1 and μ_2 are the shear and bulk viscosity, respectively, and ν is the Poisson's ratio. Additionally, $E\sqrt{\rho}$ represents Young's modulus (stiffness), which we assume to be dependent on the concentration of the collagen molecules.

Additionally, the total body force $\mathbf{f}$ in Eq. (2) needs a more precise description. We assume that the myofibroblasts generate an isotropic stress due to their pulling on the extracellular matrix, which is proportional to the product of the cell density of the myofibroblasts and a simple function of the concentration of the collagen molecules:

$$\mathbf{f} = \nabla\cdot\boldsymbol{\psi},$$

$$\boldsymbol{\psi} = \xi M\left(\frac{\rho}{R^2 + \rho^2}\right)\mathbf{I}.$$

Here, $\boldsymbol{\psi}$ is a second-order tensor representing the total generated stress by the myofibroblast population, the parameter ξ is the generated stress per unit cell density and the inverse of the unit collagen concentration, and R is a fixed constant.

Finally, we consider the growth contribution tensor $\mathbf{G}$ in Eq. (3). We assume that the rate of active change of the effective strain is proportional to the product of the amount of effective strain, the local concentration of the MMPs, the local concentration of the signaling molecule, and the inverse of the local concentration of the collagen molecules. Taken collectively, this results in the following symmetric tensor:

$$\mathbf{G} = \zeta\left(\frac{g(N,M,c,\rho)c}{\rho}\right)\boldsymbol{\varepsilon} = \zeta\left(\frac{(N+\eta^{II}M)c}{1+a_c^{III}c}\right)\boldsymbol{\varepsilon},$$

where the parameter ζ is the rate of morphoelastic change.

2.1.7. Boundary conditions

We locate the *xy*-plane parallel to the surface of the skin and write

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \end{bmatrix}.$$

In other words, the skin is assumed to be homogeneous perpendiculat (orthogonal) to the skin surface. This follows the approach of (Koppenol et al., 2017), which assumes an infinitesimally thin skin layer.

We perform computations on a reduced, symmetrical domain to reduce the computational workload. The solution inherits this symmetrical property. We define the *computational domain* by $\Omega_\mathbf{x}$ and the *boundary of the computational domain* by $\partial\Omega_\mathbf{x}$. The distance between $\partial\Omega_\mathbf{x}$ and the modelled burn is sufficiently large to prevent too much variable diffusion near the boundary. Furthermore, the computational domain implicitly depends on the time t given that $\mathbf{x} = \mathbf{x}(t)$. We do not specify any boundary conditions for ρ and $\boldsymbol{\varepsilon}$ because of over-determination, since the equations for ρ and ε are ordinary differential equations for time t.

Let $\partial\Omega_\mathbf{x} = \left\{\partial\Omega_\mathbf{x}^o, \partial\Omega_\mathbf{x}^h, \partial\Omega_\mathbf{x}^v\right\}$, where $\partial\Omega_\mathbf{x}^o$ represents the outer non-symmetrical boundaries, $\partial\Omega_\mathbf{x}^h$ represents the horizontal symmetrical boundary where $y = 0$, and $\partial\Omega_\mathbf{x}^v$ represents the vertical symmetrical boundary where $x = 0$. For the chemicals, the following boundary conditions hold for all time t and all

$$\mathbf{x} \in \partial\Omega_\mathbf{x}^o : \quad N(\mathbf{x};t) = \overline{N}, \quad M(\mathbf{x};t) = \overline{M},$$
$$\text{and} \quad c(\mathbf{x};t) = \overline{c},$$
$$\mathbf{x} \in \partial\Omega_\mathbf{x}^p : \quad \mathbf{J}_{N/M/c}\cdot\mathbf{n} = 0,$$

where $p \in \{h, v\}$, and $\mathbf{n}$ is the outward pointing normal vector. Note that $p = h$ and $p = v$, respectively, correspond to the horizontal and vertical (internal) boundary of symmetry. We use similar conditions for the mechanics, for all time t and all

$$\mathbf{x} \in \partial\Omega_\mathbf{x}^o : \quad \boldsymbol{v}(\mathbf{x};t) = 0,$$
$$\mathbf{x} \in \partial\Omega_\mathbf{x}^p : \quad \boldsymbol{v}\cdot\mathbf{n} = 0 \quad \text{and} \quad (\sigma\cdot\mathbf{n})\cdot\tau = 0,$$

where τ is the tangential vector. The boundary conditions reflect that there is no flux through the symmetry boundary. The outer boundary is assumed to be sufficiently far away from the wound in the sense that, far away, the skin is undamaged. Hence, the densities of the (myo)fibroblasts reflect the undamaged state there. Symmetry implies that the normal component of the displacement velocity is zero. Furthermore, the tangential component of the velocity at the symmetry boundary is not necessarily zero, but the shear stress is zero. Since the strain ε equation does not involve any spatial partial derivatives of ε, boundary conditions are not to be imposed for ε. The above boundary conditions, combined with initial conditions for all the quantities, are necessary and sufficient for mathematical well-posedness of the problem in the sense of the existence of a uniquely defined solution. The proof is beyond the scope of the paper.

2.1.8. Initial conditions

The initial conditions describe the cell densities and concentrations at the onset of the proliferative phase of wound healing. Signaling molecules are present in the wound because they are secreted in the inflammatory phase of wound healing. Furthermore, fibroblasts and collagen are initially assumed to be present in the wound, whereas myofibroblasts are assumed to be absent. The initial wound geometry is defined by the initial profiles of the collagen, (myo)fibroblasts and concentration of the chemokine.

The initial wounded area is denoted by $\Omega^w(0) \subset \Omega_{\mathbf{x},0}$. The unwounded area is then $\Omega_{\mathbf{x},0}\backslash\Omega^w(0)$. Let $d(\mathbf{x})$ be the shortest distance from a point $\mathbf{x} \in \Omega^w$ to the wound boundary. Let $\Omega_s^w = \{\mathbf{x} \in \Omega^w(0) : d(\mathbf{x}) \geq s\}$. Then, for $z \in \{N, c, \rho\}$ we have the following initial densities or concentrations:

$$z(\mathbf{x}, 0) = \begin{cases} \widetilde{z}, & \mathbf{x} \in \Omega_s^w, \\ \overline{z}, & \mathbf{x} \in \Omega_{\mathbf{x},0}\backslash\Omega^w(0), \end{cases}$$

where $\widetilde{z}, \overline{z} \in \mathbb{R}^+$, the latter indicating the equilibrium value. Furthermore,

$$M(\mathbf{x}, t) = \overline{M} = 0, \quad \mathbf{x} \in \Omega_{\mathbf{x},0}.$$

For the wound boundary steepness, we use half a period of sine-functions for N, c, and ρ to smoothly transition from the wound to the unwounded area.

Regarding the initial conditions for the mechanical part of the model, all quantities are initialized with zero.

2.1.9. Numerical discretizations and solvers

We solve the model equations by applying the finite element method. We note that we add the term $\varepsilon_{ij}[\nabla\cdot\boldsymbol{v}]$ for $i, j \in \{1, 2\}$ to the left-hand side and the right-hand side of the effective Eulerian strain equations before deriving the weak formulation. Furthermore, we make use of $\varepsilon_{21} = \varepsilon_{12}$ (symmetric effective strain tensor), which we have proved for all time t (Egberts et al., 2020). For a complete overview of the derivation of the finite element formulations, we refer to the appendix in Koppenol's PhD thesis (Koppenol, 2017).

We subdivide the computational domain into a finite number of non-overlapping triangles Δ_p (i.e., the elements) that are as equilateral as

possible. Let $X_h(t) \subset H^1(\Omega_{\mathbf{x}})$ the finite element subspace and $\mathbf{a}_j$, $j \in \{1, \ldots, n\}$, $n \in \mathbb{N}$ the coordinates of the vertices of the elements. We choose piecewise linear Lagrangian basis functions $\varphi_i \in X_h(t)$ with $\varphi_i(\mathbf{a}_j; t) = \delta_{ij}$ for $i, j \in \{1, \ldots, n\}$ as basis functions for the finite-dimensional subspace $X_h(t)$, where δ_{ij} denotes the Kronecker delta function.

We approximate the local displacements of the dermal layer ($\mathbf{u}$) (with)

$$\mathbf{u}(\mathbf{x}; t + \Delta t) \approx \mathbf{u}(\mathbf{x}; t) + \mathbf{v}(\mathbf{x}; t)\Delta t.$$

We use the initial condition $\mathbf{u}(\mathbf{x}; 0) = 0$, $\forall \mathbf{x} \in \Omega_{\mathbf{x},0}$. Furthermore, we update the mesh (triangulation) in every time integration step. For the sake of obtaining reliable numerical solutions with a 'well-conditioned' discretization (where the impact of rounding errors is negligible), it is crucially important that the mesh quality is good. A high-quality mesh requires that the triangular elements should be as close as possible to equilateral triangles. The change of shape of the computational domain near the scar makes it important to be aware of the mesh quality. In order to warrant a good mesh, several quantifiers that assess the mesh quality exist. We determine the quality of this updated mesh by computing

$$\min_{e_k} |\mathbf{J}_{e_k}| \Big/ \max_{e_k} |\mathbf{J}_{e_k}|, \quad e_k \in \Omega,$$

with $\mathbf{J}$ the Jacobian for the element using a positively oriented numbering to warrant positiveness of the Jacobians. In case $\min_{e_k} |\mathbf{J}_{e_k}| / \max_{e_k} |\mathbf{J}_{e_k}| < 0.5$, we perform (global) remeshing. The absolute value of the determinant of the Jacobian is twice the area of a triangular element. Elements that are near degenerate, that is, having a large aspect ratio, tend to have at least one very small angle (and possible one very large angle), which makes the discretization prone to violating the DeLaunay criterion (the sum of opposite angles of two triangles that share a side is at most π_s). Violation of this criterion will also make the finite element stiffness matrix, as a discrete counterpart of the Laplace operator, no longer be an M-matrix (Xiao et al., 2018), thereby no longer warranting properties like monotonicity, discrete maximum principles and it is no longer guaranteed that spurious oscillations are avoided. Since high aspect ratios of elements are connected to relatively low areas, this hints at having the above ratio close to one, avoiding these problems.

We use mass lumping and a semi-implicit flux corrected transport limiter (Möller et al., 2008) that enforces the positiveness of solutions so that loss of monotonicity (i.e., spurious oscillations) is suppressed.

2.1.10. Relative surface area

The wound boundary is explicitly tracked throughout the simulation. Initially, a vector of 100 nodes represents the wound boundary. At each time step, this boundary is updated using interpolated velocity displacements, ensuring that points on the boundary remain on the boundary. Having these boundary points, allows the computation of the wound area at each time.

During wound healing, due to myofibroblasts pulling on the surrounding collagen fibers, the wound contracts towards its center and retracts after these cells disappear. *The relative surface area* (of the) *wound* (RSAW) is an important measure, as it gives valuable information about contraction. It is defined as follows:

$$RSAW(t) = \frac{area(\Omega^w(t))}{area(\Omega^w(0))}.$$

The minimum RSAW value corresponds to maximum contraction, and the asymptotic value indicates the long-term (permanent) contraction.

2.2. Machine learning models

In this section, we introduce the methodology of our machine learning models for predicting the evolution of the wound shape over time. Therefore, we first provide a brief introduction to neural networks, then introduce the concept of operator learning, and finally, introduce the network architecture of our machine learning model, which is based on DeepONets (Lu et al., 2021).

2.2.1. Neural networks

Let us consider a generic supervised machine learning task. Therefore, let input data $X = (x_1, \ldots, x_N)$ and output data $Y = (y_1, \ldots, y_N)$ be given, with $x_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^m$, for $1 \leq i \leq N$. Then, we aim at finding a parametrized function

$$f_\theta : \mathbb{R}^n \to \mathbb{R}^m,$$

such that

$$f_\theta(x_i) \approx y_i; \tag{4}$$

where $\theta \in \mathbb{R}^K$ is a vector consisting of all parameters of the model.

Neural networks are a specific class of parametrized functions. In particular, neural networks are composed of functions of the form

$$f_{W_j, b_j}^{(j)}(x) := \sigma(W_j \cdot x + b_j) \tag{5}$$

with the index j, $W_j \in \mathbb{R}^{n_j \times n_{j+1}}$ and $b_j \in \mathbb{R}^{n_{j+1}}$, and a nonlinear *activation function* σ which is applied component-wise to a vector; typical examples for the activation function σ are shown in Fig. 1(a). Here, we consider only $\sigma(x) = \max(0, x)$, (ReLU). The function $f_{W_j, b_j}^{(j)}$ is a composition of a multivariate affine function and a nonpolynomial activation function σ. A simple feedforward neural network, or *multi-layer perceptron (MLP)*, is then the composition of multiple functions of the form given in Eq. (5) and one final linear map, that is,

$$f_\theta := A \circ f_{W_l, b_l}^{(l)} \circ f_{W_{l-1}, b_{l-1}}^{(l-1)} \circ \cdots \circ f_{W_1, b_1}^{(1)}.$$

Here, $A \in \mathbb{R}^{m \times n_{l+1}}$ corresponds to the linear *output layer*, and each function f_j, for $1 \leq j \leq l$, corresponds to one so-called *hidden layer* of the model; see also Fig. 1(b). The matrices and vectors W_j and b_j, for $1 \leq j \leq l$, as well as the matrix A constitute the parameter vector θ. We call a neural network *deep* if the number of hidden layers is sufficiently large, typically for at least 3 hidden layers.

In order for the model f_θ to fit the data as in Eq. (4), a minimization problem is formulated. In the simplest case of regression, fitting the model can be equivalently written as finding

$$\theta^\star = \arg\min_\theta \mathscr{L}(\theta) \quad \text{with} \quad \mathscr{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \| f_\theta(x_i) - y_i \|_2^2.$$

We also call $\mathscr{L}(\theta)$ the *loss function*, which here is based on the mean squared error (MSE). Due to the nonlinearity of f_θ, the loss function is complicated and, in general, non-convex. The minimization is usually performed using a gradient-based optimizer or using a quasi-Newton method. Typical examples are stochastic gradient descent (SGD), and its variant Adam (Kingma and Ba, 2014), or the limited memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method (Liu and Nocedal, 1989). These methods require gradient information, which can be computed efficiently using automatic differentiation using the back-propagation algorithm (Kelley, 2012).

For more details, see for instance (Goodfellow et al., 2016). Note that neural networks can also be used for semi-supervised and unsupervised learning. However, we will only consider the case of supervised learning in this paper.

2.2.2. Operator learning

In (Kovachki et al., 2021), neural operators have been introduced as
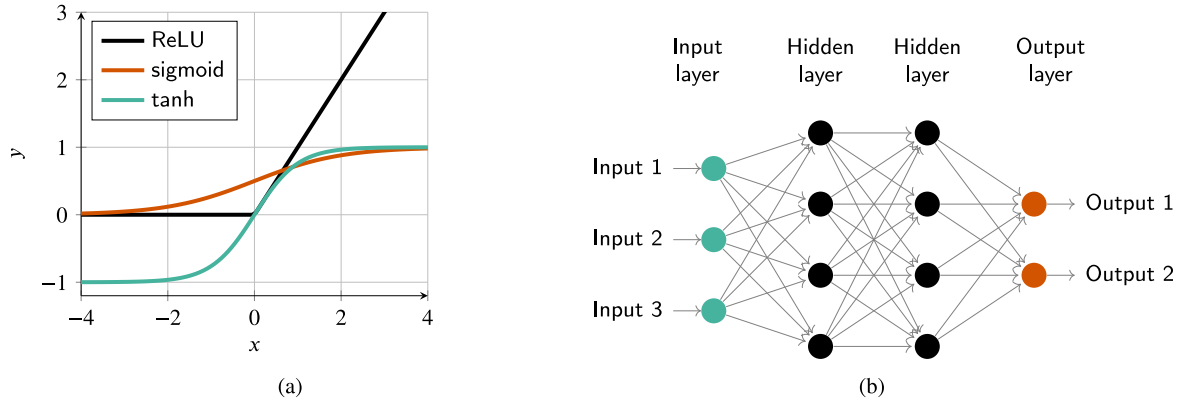
Fig. 1. Nonlinear activation functions (a) and a feedforward neural network with two hidden layers (b).

neural networks for approximating operators, that is, maps between infinite-dimensional function spaces. Suitable neural operator architectures have universal approximation properties, meaning that, with sufficient model capacity (i.e., a sufficiently large number of model parameters), they can approximate any given nonlinear continuous operator up to arbitrary precision. Furthermore, in (Kovachki et al., 2021), it is said that neural operators can be made discretization-invariant for the following reasons: 1) the model can act on any discretization of the input function; 2) the model can be evaluated at any point in the output domain; 3) when refining the discretization, the model converges to a continuum operator.

Neural operators are particularly well-suited for creating surrogate models for numerical solvers for PDEs, since the input of the model can be a function, such as

- a right-hand side function of a PDE,
- a spatial distribution of a material parameter,
- a function describing a boundary condition, or
- a level-set function describing the geometry,

and the output of the neural operator can be the solution of the initial boundary value problem (IBVP). This is different from, for instance, classical physics-informed neural networks (PINNs) (Raissi et al., 2019), which generally only approximate the solution of a single IBVP.

Recently, many different neural operator architectures have been introduced, with Fourier neural operators (FNOs) (Li et al., 2020) and DeepONets (Lu et al., 2021) being the most popular choices. We also highlight convolutional neural operators (CNOs) (Raonić et al., 2023), which satisfy structure-preserving continuous-discrete equivalence, enabling learning operators without discretization-dependent aliasing errors. The proposed CNO architecture is an extension of the popular U-Net architecture (Ronneberger et al., 2015).

Here, we build a neural operator learning model based on the DeepONet architecture, and for the sake of brevity, we will focus only on a brief description of this approach. To this extent, consider a generic nonlinear continuous operator $G : V \to W$, where V and W are compact, nonempty subsets of Banach spaces, which amount to normed function spaces on a nonempty, compact subset of physical space-time. We are interested in evaluating $G(v)$ at discrete points in the nonempty compact portion of physical space-time. The user has to input two sets: (1) the points in a compact region in space-time where (s)he wants to evaluate $G(v)$ and (2) the input function at points in a compact region in space-time, that is $v(x_1), \dots, v(x_m)$. Note that the compact regions where v is obtained and where $G(v)$ is determined do not have to be the same. The Universal Approximation Theorem for Operator Learning states that each nonlinear mapping between nonempty subsets of Banach spaces can be approximated arbitrarily well, provided that sufficient input information (such as the dimensionality and features of the input data)

and complexity (the depth, width, and structure of the neural network) are incorporated:

Theorem 1. (Universal approximation for operators (Lu et al., 2021; Chen and Chen, 1995)).

Let σ be a continuous nonpolynomial function, X be a Banach space, $K_1 \subset X$ and $K_2 \subset \mathbb{R}^d$ two compact subsets, respectively, V be a compact subset of $\mathscr{C}(K_1)$, and G be a nonlinear continuous operator $V \to \mathscr{C}(K_2)$. Then, for any $\varepsilon > 0$, there exist positive integers n, p, m and constants c_i^k, $w_{ij}^k, b_i^k, \widehat{b}_k \in \mathbb{R}, \widehat{w}_k \in \mathbb{R}^d, x_j \in K_1$, for $i = 1, \dots, n$, $k = 1, \dots, p$, and $j = 1, \dots, m$, such that

$$\left| G(v)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma \left(\sum_{j=1}^{m} w_{ij}^k u(x_j) + b_i^k \right)}_{branch} \underbrace{\sigma(\widehat{w}_k y + \widehat{b}_k)}_{trunk} \right| < \varepsilon \qquad (6)$$

for all $v \in V$ and $y \in K_2$.

The idea of DeepONets is then to replace what is denoted as *branch* and *trunk* in Eq. (6) by (deep) neural networks. Note that Eq. (6) relates to Eq. (5) in the sense that w_{ij}^k and $\widehat{w}_k$ are the coefficients of weight matrices W_j of the branch and trunk, and b_i^k and $\widehat{b}_k$ are the coefficients of bias vectors b_j. Section 2.2 warrants the existence of such a network; however, it says nothing about the rate of convergence.

In its general form, the DeepONet architecture is depicted schematically in Fig. 2. Intuitively, the trunk network corresponds to spatial basis functions $t_1, \dots, t_p$, whereas the branch network provides corresponding coefficients $b_1, \dots, b_p$ in the basis representation

$$G(v)(y) \approx \sum_{k=1}^{p} b_k(x_1, \dots, x_m) t_k(y).$$

Note that DeepONet models are generally not discretization-invariant since they depend on a fixed choice of sensor locations or basis functions used to discretize the input function. This has been
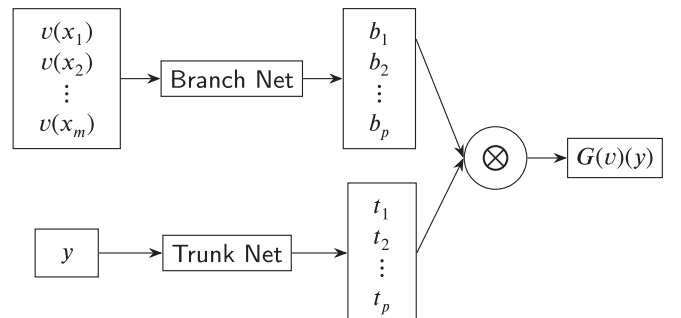

Fig. 2. (Unstacked) DeepONet architecture; based on (Lu et al., 2021).

observed in literature, and some methods have been proposed to make DeepONets invariant to input function discretization; see, for instance, (Lin et al., 2023; Zhang et al., 2023; Zhang et al., 2024; Zhang et al., 2025). Many other variants of DeepONets have been introduced, where specific types of networks are chosen as the branch and trunk networks, for instance, U-Net (Diab and Al-Kobaisi, 2023) and graph neural network (GNN) (Sun et al., 2022) architectures. In many related works, the basis functions are prescribed instead of learning them via a neural network, for instance, using classical discretizations (Griese et al., 2024; Meethal et al., 2023; Möller et al., 2021; Wang et al., 2020). Furthermore, the basis property has also been recently used to learn coarse basis functions for robust preconditioning (Kopaničáková and Karniadakis, 2024). For improved precision, DeepONets can be stacked (Howard et al., 2023) and combined with domain decomposition techniques (Heinlein et al., 2024).

2.2.3. Deep operator network architecture

We propose a machine learning model based on the DeepONet architecture for predicting post-burn contraction over time. The exact architecture of our model is schematically depicted in Fig. 3. It consists of a branch and trunk network, both of which are simple MLPs with three hidden layers, containing 50 neurons per layer. The branch takes as input five parameters from the morphoelastic model describing wound contraction: the cell diffusion constant D_F, the chemotactic constant χ_F, the signaling diffusion constant D_c, the cell differentiation rate k_F, and the half-maximum cell division enhancement rate a_c^I. These parameters describe the behavior of cells involved in the wound healing process and are characteristics that vary between patients. Note that, in our approach, the input to the branch is not a discretization of a sampled function from a chosen function space (as depicted in Fig. 2). Instead, we have an exact parametrization of the problem, where the five parameters are assumed to be constant across space-time. The branch network outputs a vector $(b_1, \ldots, b_{2p}) \in \mathbb{R}^{2p}$

The trunk network takes as input a coordinate (t, x, y) in which we want to evaluate the displacement $\mathbf{u}(t, x, y) = (u_1, u_2)$. The trunk additionally takes as input information about the initial wound shape, given by the quadruple $(y_{cut}, x_m, y_m, x_{cut})$. Fig. 4 shows the three wound shapes used for training (see also Section 3.1) and visualizes this extra input to the trunk. The latter uniquely determines the initial wound shape at $t = 0$. The latter uniquely determines the initial wound shape at $t = 0$. The

coordinate (x_m, y_m) is chosen as follows: for each shape, the boundary is discretized into 100 spatial points. These points are ordered, starting from $(0, y_{cut})$ at index $i = 1$, up to $(x_{cut}, 0)$ at index $i = 100$. For the rectangle, we select for (x_m, y_m) the unique corner point. For the ellipse and rhombus, we take the point at index $i = 50$. While this point is near the geometric midpoint, it does not necessarily correspond to it, due to non-equidistant spacing of the points. The output of the trunk network is a vector $(c_1, \ldots, c_p) \in \mathbb{R}^p$. In our simulations, we fix $p = 50$.

The output of the DeepONet is an inner product of the outputs of the trunk and branch networks. Specifically, let $T(\cdot)$ be the operation of the trunk and $B(\cdot)$ be the operation of the branch network, then we have

$$c_1, \ldots c_p = T(t, x, y, y_{cut}, x_m, y_m, x_{cut}),$$

$$b_1, \ldots b_{2p} = B(D_F, \chi_F, D_C, k_F, a_C^I),$$

$$\widehat{u}_1 = \sum_{i=1}^p b_i c_i,$$

$$\widehat{u}_2 = \sum_{i=1}^p b_{i+p} c_i.$$

Finally, we apply *sine augmentation* to the DeepONet outputs, ensuring zero displacements at the top and right boundaries of our domain. Moreover, it ensures zero x-displacement on the left boundary and zero y-displacement on the bottom boundary. Fig. 5(a) summarizes the effect on all four boundaries. This hard enforcement of boundary conditions is not new; examples can already be found in early works employing neural networks to solve PDEs; see, for instance, (Lagaris et al., 1998). The final outputs of our model are given by

$$\begin{aligned} u_1 &= \widehat{u}_1 \cdot \sin\left(\frac{\pi}{x_l} x\right) \cos\left(\frac{\pi}{2y_l} y\right), \\ u_2 &= \widehat{u}_2 \cdot \sin\left(\frac{\pi}{y_l} y\right) \cos\left(\frac{\pi}{2x_l} x\right). \end{aligned} \tag{7}$$

The size of the domain, described by (x_l, y_l), is visualized in Fig. 4 as well.

3. Datasets and training

This section describes the approaches we used to generate datasets to train and test our model and the specific training setup.
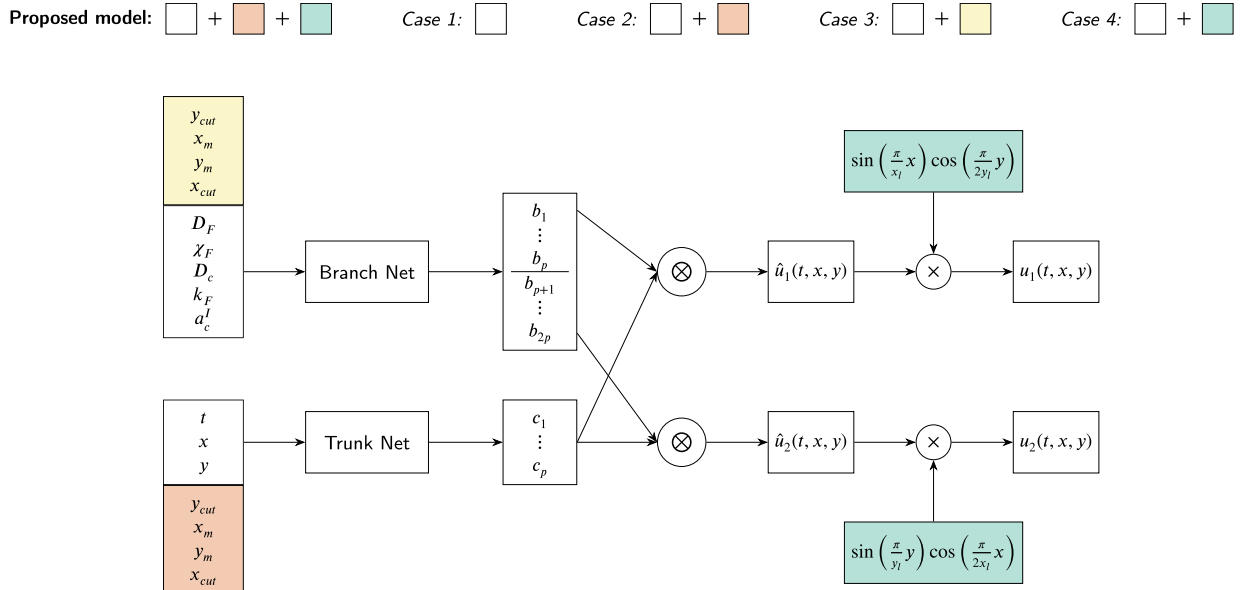

Fig. 3. The DeepONet architecture used for predicting dermal displacement. The legend at the top describes four other cases against which we have tested our proposed model; see Table 2.
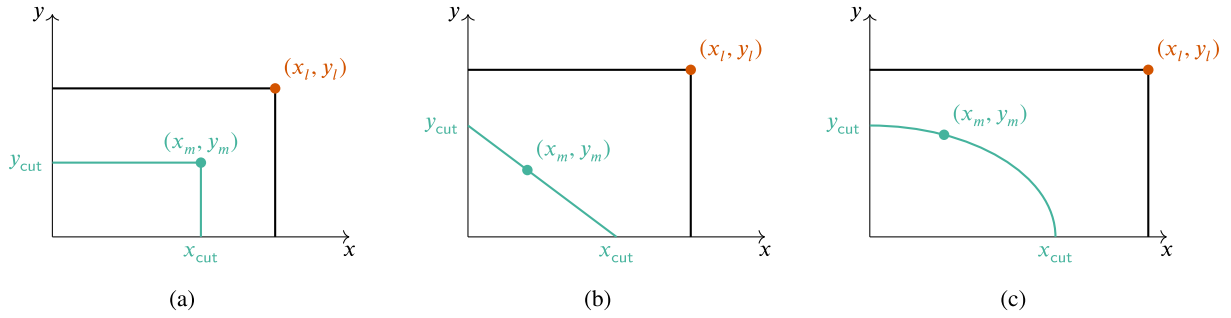
Fig. 4. The three initial wound shapes used for training: rectangle (a), rhombus (b), and ellipse (c). Due to symmetry, a quarter of the complete domain is considered. Additionally, a visualization of the quadruple $(y_{cut}, x_m, y_m, x_{cut})$, which serves as extra input to the trunk. The coordinate (x_l, y_l) is required for the sine augmentation step.
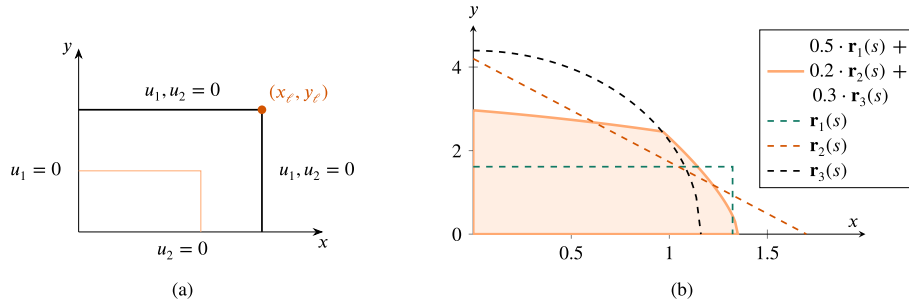

Fig. 5. Enforcement of boundary conditions as a result of sine augmentation (a) and a convex combination of the three basic wound shapes (b).

Table 1

Ranges for the values of the five parameters taken as input to the branch network.

Parameter	Range	Dimension
D_F	$7.6167 \cdot 10^{-7} - 1.2 \cdot 10^{-6}$	cm ^{5}/(cells day)
χ_F	$(2-3) \cdot 10^{-3}$	cm ^{5}/(g day)
D_c	$(2.22 - 3.2) \cdot 10^{-3}$	cm ^{2}/day
k_F	$8 \cdot 10^6 - 1.08 \cdot 10^7$	cm ^{3}/(g day)
a_c^I	$(0.9 - 1.1) \cdot 10^{-8}$	g/cm ^{3}

3.1. Training set

We generate a training set consisting of three different geometries for the initial wound shape: a rectangle (including a square), a rhombus (including a rotated square), and an ellipse (including a circle). We consider a quarter of the complete domain, assuming symmetry along the x- and y-axis, as shown in Fig. 4.

We run 750 finite element simulations, each run uniformly selecting one of the three initial wound shapes. For the wound size, we uniformly sample $x_{cut}, y_{cut} \in (0, 5)$ centimeters. The size of the complete domain is then determined as

$$x_l = 2.5 \cdot x_{cut},$$
$$y_l = 2.5 \cdot y_{cut}, \qquad (8)$$

rounded to one decimal. For all simulations, we fix $t_{end} = 100$ days. The training data sampling strategy is as follows: for each finite element simulation, 10 time steps are uniformly sampled. Then, for each time step, 20 spatial coordinates from the domain are uniformly sampled. This gives us 200 coordinates (t, x, y) per simulation, in which we record the displacements (u_1, u_2). Each triplet (t, x, y) is extended with the corresponding quadruple, completing the trunk network input.

For each numerical simulation, the five parameter values taken as input to the branch are sampled from uniform distributions with pre-specified ranges, as listed in Table 1. Furthermore, the values (x_l, y_l),

as given in Eq. (8), are tabulated, as they are necessary for defining the sine augmentation; cf. Eq. (7).

Following the procedure described above, each finite element simulation contributes the following quadruple to the training data:

$$
\left[
\begin{bmatrix}
D_F, \chi_F, D_c, k_F, a_c \\
D_F, \chi_F, D_c, k_F, a_c \\
\vdots \\
D_F, \chi_F, D_c, k_F, a_c
\end{bmatrix},
\begin{bmatrix}
t_1, x_1, y_1, y_{cut}, x_m, y_m, x_{cut} \\
\vdots \\
t_1, x_{20}, y_{20}, y_{cut}, x_m, y_m, x_{cut} \\
\vdots \\
t_{10}, \widehat{x}_1, \widehat{y}_1, y_{cut}, x_m, y_m, x_{cut} \\
\vdots \\
t_{10}, \widehat{x}_{20}, \widehat{y}_{20}, y_{cut}, x_m, y_m, x_{cut}
\end{bmatrix},
\right.
$$

$$
\left.
\begin{bmatrix}
u_1(t_1, x_1, y_1), u_2(t_1, x_1, y_1) \\
\vdots \\
u_1(t_1, x_{20}, y_{20}), u_2(t_1, x_{20}, y_{20}) \\
\vdots \\
u_1(t_{10}, \widehat{x}_1, \widehat{y}_1), u_2(t_{10}, \widehat{x}_1, \widehat{y}_1) \\
\vdots \\
u_1(t_{10}, \widehat{x}_{20}, \widehat{y}_{20}), u_2(t_{10}, \widehat{x}_{20}, \widehat{y}_{20})
\end{bmatrix},
\begin{bmatrix}
x_l, y_l \\
x_l, y_l \\
\vdots \\
x_l, y_l
\end{bmatrix}
\right].
$$

The first column is input to the branch, the second column is input to the trunk, the third column represents the target values, and the final column evaluates the sines and cosines in the sine augmentation step. As we perform 750 such numerical simulations, we obtain a training set containing 150 000 data points. We use an $80\% - 20\%$ split for training and validation, respectively.

3.2. Training setup

Hyperparameter tuning was initially performed on a simplified model considering only one wound shape, and the resulting parameters are applied in this case as well (for more details, see the master's thesis
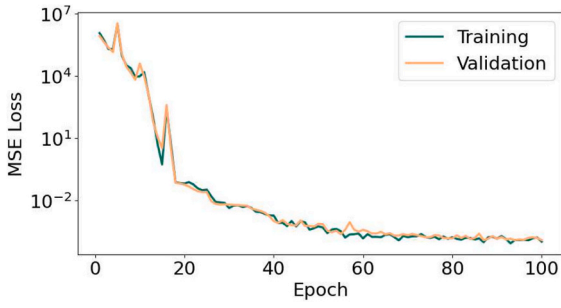
Fig. 6. Training and validation losses of the proposed DeepONet. The dataset is based on 750 finite element simulations, using an 80%–20% split for training and validation, respectively.

(Husanovic, 2024)). For training our model, we use the Adam algorithm with standard backpropagation to minimize the MSE loss. The learning rate is set to 0.001. Through empirical testing, we found that a batch size of 100 yields satisfactory results. We train the network for 100 epochs, after which we observe that the loss stagnates around 10^{-3}. Training the model for more epochs does not further decrease the loss. Fig. 6 shows a plot of the training and validation losses.

3.3. Convex test set

We test the performance of our model on what we call the *convex test set*. This is a more generic data set, containing convex combinations of the three basic shapes used for training. Let $s \in [a,b]$ and $\mathbf{r}_i = \mathbf{r}_i(s)$ represent a parametrization of the (piecewise) smooth boundary of basic wound shape i ($i = 1$ corresponding to a rectangle, $i = 2$ to a rhombus, and $i = 3$ to an ellipse), on which the DeepONet is trained. Then, a convex combination of the three parametrizations is given by

$$\mathbf{r}(s, \alpha_1, \alpha_2, \alpha_3) = \sum_{p=1}^{3} \mathbf{r}_p(s) \alpha_p, \qquad (9)$$

where $\alpha_p \geq 0$ and $\sum_{p=1}^{3} \alpha_p = 1$. This parametrization provides the boundary of a new wound shape. Fig. 5(b) gives a visualization.

The convex test set was created by executing 150 finite element simulations. For each simulation, we create a convex combination of the three basic shapes. To this end, we first fix the sizes of the basic shapes by uniformly choosing $x_{cut}, y_{cut} \in (0,5)$ centimeters. Then, we randomly select three weights that sum up to one and perform the multiplication as given in Eq. (9).
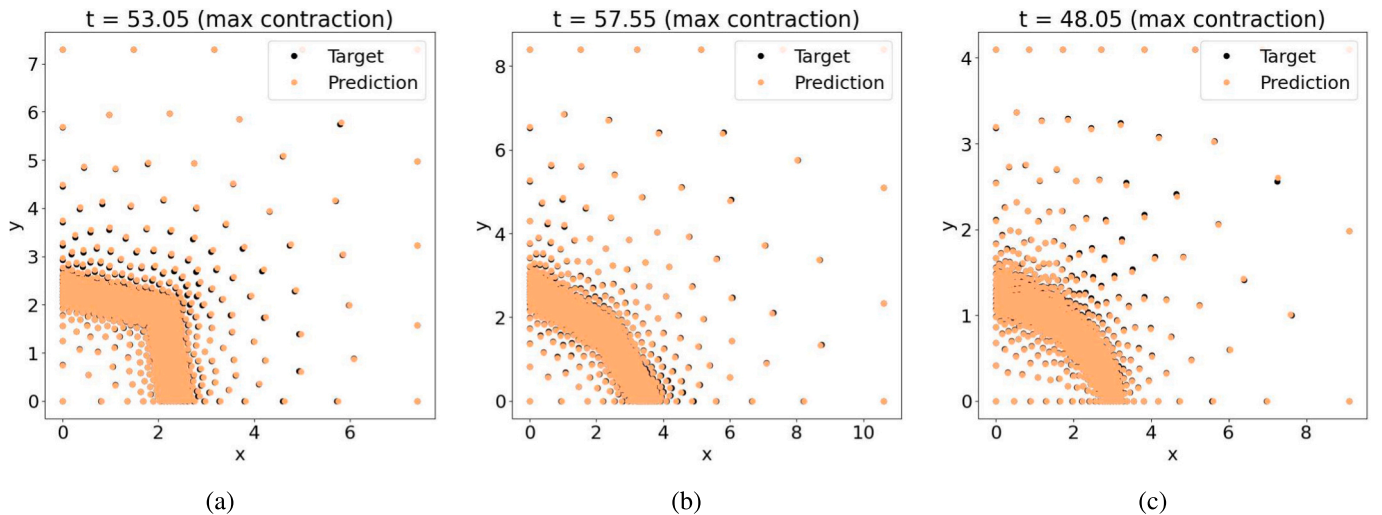
The remainder of the data sampling strategy is similar to the one used for generating the training data described in Section 3.1. The only difference now is that for each finite element run, we sample all available time steps, and for each time step, we sample all available spatial coordinates in the domain. This procedure results in a convex test set containing 18,035,821 data points.

We note that the quadruples $(y_{cut}, x_m, y_m, x_{cut})$ no longer uniquely define the initial wound shapes for this dataset. This is because it contains convex combinations of the three basic shapes, which do not have a straightforward geometric definition that can be captured by these four values alone. As a result, the network does not know beforehand the exact initial shapes of the samples in the convex test set.

4. Results

This section presents the results. We begin by examining the performance of the trained DeepONet on the convex test set, followed by a comparison to four other architectural setups. Lastly, we assess predictions over an extended period of one year.

4.1. Performance on convex test set

Our DeepONet predicts the x and y displacements in any given spatial coordinate in the domain at any given time. We evaluate the trained model on each sample in the convex test set. Fig. 7 shows three different examples of the obtained predictions versus the corresponding targets, each at its respective time of maximal contraction. Note that we depict the actual coordinates after applying the displacements here, rather than the displacements. We observe good prediction abilities, especially at the wound boundary. Given that most of the spatial coordinates in the training set are located at the wound boundaries, the network is expected to perform best in these regions. Further away from the wound, the prediction accuracy decreases slightly. As we move towards the top and right boundary of the domain, the displacements go to zero. We observe this in our predictions as well, which was ensured by the sine augmentation step during training.

Furthermore, given that our model is trained on the three basic wound shapes, Fig. 7 gives an indication of good generalization abilities. This is further validated in the scatter plots in Fig. 8, plotting the true displacements against the predicted displacements. The line $y = x$ indicates perfect predictions of the neural network model. The scatter plots contain all convex test set samples at all time steps, that is, around 18 million data points. We observe a strong concentration around the lines $y = x$, with no unusual outliers. This indicates that our model gives
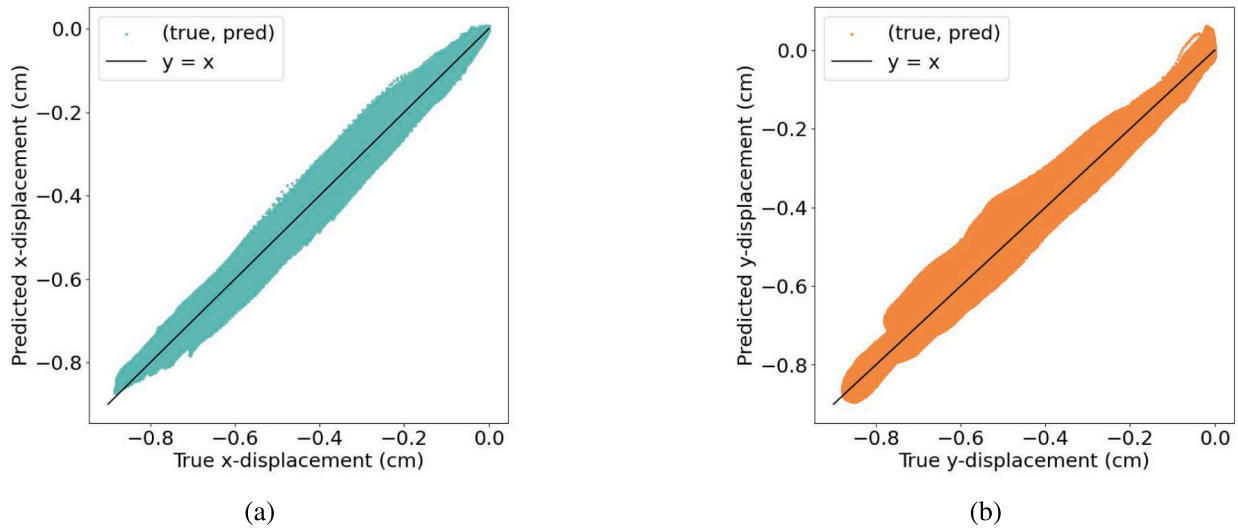

Fig. 7. Prediction vs. target for three samples from the convex test set, at their respective time of maximal contraction.

Fig. 8. True vs. predicted displacement in *x*- (a) and *y*-direction (b) on the convex test set.

good predictions and that it generalizes well to convex combinations of the shapes is was trained on. We note that the scatter plots for *x*- and *y*-displacement are not exactly the same, as no symmetry was enforced; consequently the performance also differs slightly. For both *x*- and *y*-displacement, we observe that the model is best at predicting very small magnitude displacements (close to zero) and the largest magnitude displacements (around −0.8 cm). The former occurs near the boundary of the domain at all time steps, and at all spatial coordinates when *t* is small. The latter occurs at the wound boundary at maximal contraction, for the largest wound sizes present in the convex set.

Additionally, we evaluate our trained model on the wound boundary. We consider the same convex test set, but only the spatial coordinates on the wound boundary are taken. This enables us to determine the RSAW at each time step. Figs. Fig. 9(a) and (b) show the best and worst predictions in terms of the RSAW on the convex set along with their initial wound geometries. In the best case, the RSAW curve mostly overlaps the target, with a very slight underestimation of the maximal contraction and overestimation of the final value at *t* = 100 days. In the worst case, the model predicts an earlier maximal contraction (49 days vs. 53) and retraction, with an overestimation of around 4% less contraction from *t* = 55 days and onwards.

Fig. 9(c) gives more insight into the behavior of the error over time.

Here, the mean and standard deviation of the absolute error over the convex test set are depicted. We observe that the absolute error increases over time. The model is most accurate in the first 18 days, where the error is up to 0.5% on average. After this time, the error, and the uncertainty thereof, start to increase. This coincides with a very steep contraction. A notable peak can be observed around day 38, after which there is a local minimum (of 1.2% on average) around day 50. The latter coincides with the average time of maximal contraction, indicating that the model is better at predicting the minimum RSAW (maximal displacement). The highest absolute error, with the largest uncertainty, is around day 90. For all times, the mean plus standard deviation of the error never exceeds 3%. Since the model predicts displacements within the rough range of [−0.9, 0] centimeters, a 3% error corresponds to deviations on the order of tenths of millimeters.

4.2. Comparison to different architectures

Table 2 presents the performance metrics of the DeepONet, where a comparison is made with four other architecture setups. These are variations of our proposed model (see Fig. 3). First, we define the performance measures used to determine the metrics in Table 2. Subsequently, we interpret the results.
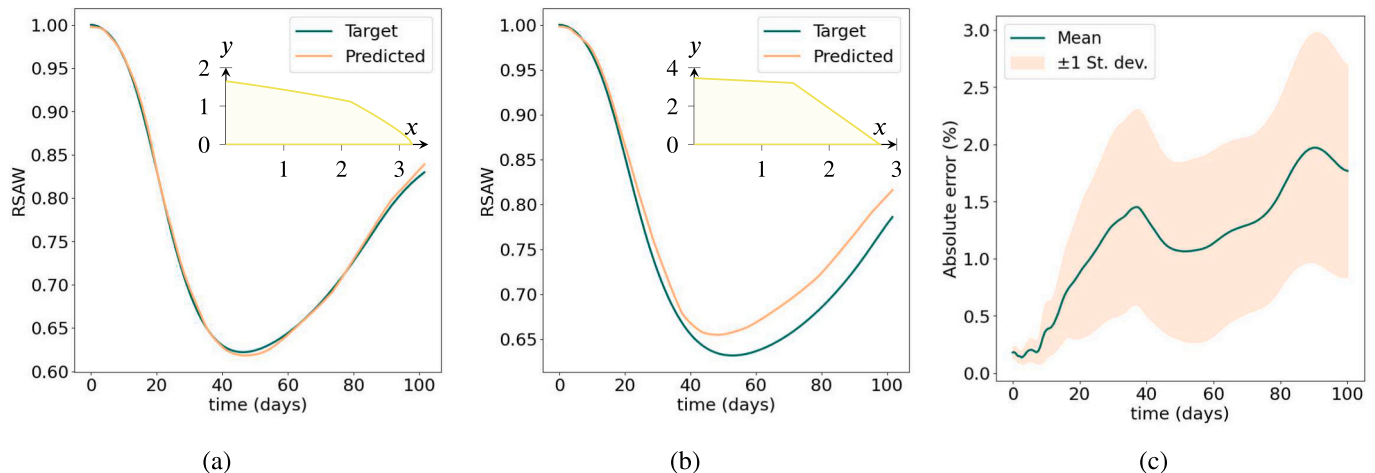

Fig. 9. Best (a) and worst (b) prediction with their corresponding geometries on the convex test set in terms of the RSAW. Mean and standard deviation of absolute error as a function of time (c). The mean is taken over all 150 samples in the convex test set.

Table 2
Performance of the proposed DeepONet compared to four other setups (see Fig. 3 for a visualization). Boldface indicates the best performance.

Performance metric	No sine aug. and no initial shape info (Case 1)	Initial shape info to branch (Case 2)	Initial shape info to trunk (Case 3)	Sine augmentation (Case 4)	Sine aug. and initial shape info to trunk (Proposed model)
R^2	0.90040	0.89613	0.99155	0.89083	**0.99437**
aRRMSE	0.31560	0.32229	0.09166	0.33032	**0.07497**
aRelErr	0.12160	0.11673	0.04432	0.15050	**0.03429**

4.2.1. Performance measures

We include the goodness of fit (R^2) statistic, the average relative root mean squared error (aRRMSE), and the average relative error (aRelErr):

$$R^2 = \frac{1}{2} \sum_{t=1}^{2} \left[1 - \frac{\sum_{i=1}^{n} \left(u_t^{(i)} - \widehat{u}_t^{(i)} \right)^2}{\sum_{i=1}^{n} \left(u_t^{(i)} - \overline{u}_t \right)^2} \right], \tag{10}$$

$$aRRMSE = \frac{1}{2} \sum_{t=1}^{2} \sqrt{\frac{\sum_{i=1}^{n} \left(u_t^{(i)} - \widehat{u}_t^{(i)} \right)^2}{\sum_{i=1}^{n} \left(u_t^{(i)} - \overline{u}_t \right)^2}}, \tag{11}$$

$$aRelErr = \frac{1}{2} \sum_{t=1}^{2} \frac{1}{n_t} \sum_{i=1}^{n} \mathbb{I}\left(u_t^{(i)} \neq 0 \right) \cdot \left| \frac{u_t^{(i)} - \widehat{u}_t^{(i)}}{u_t^{(i)}} \right|, \tag{12}$$

where

- $u_t^{(i)}$ is the true displacement (rounded to one decimal place in Eq. (12)),
- $\widehat{u}_t^{(i)}$ is the predicted displacement (rounded to one decimal place in Eq. (12)),
- $\overline{u}_t$ is the average true displacement,
- $\mathbb{I}\left(u_t^{(i)} \neq 0 \right)$ is the indicator function, which is 1 if $u_t^{(i)} \neq 0$ and zero otherwise,
- n_t is the number of samples where $u_t^{(i)} \neq 0$,
- n is the total number of samples.

A value of $R^2 = 1$ indicates perfect predictions, whereas $R^2 = 0$ indicates that the model is not better than always returning the expected values. For our proposed DeepONet, we find $R^2 = 0.99$. This shows that the model can accurately predict dermal displacement. Furthermore, we find an aRRMSE of 0.08, showing excellent performance according to Despotovic et al. (Despotovic et al., 2016). Lastly, the aRelErr is found to be only 0.03, which further indicates that the model can excellently reproduce the finite element simulations.

We compare the proposed DeepONet architecture to four variations thereof. Fig. 3 shows which blocks are included in the different cases, corresponding to the columns of Table 2. We observe that, compared to all variations, the proposed model gives the best performance. The "skeleton" DeepONet, where the model does not have any information about the initial wound shape and no sine augmentation is applied (case 1), performs significantly worse. We find $R^2 = 0.90$, aRRMSE = 0.32, and aRelErr = 0.12. Adding initial wound shape info in the form of the quadruple $(y_{cut}, x_m, y_m, x_{cut})$ to the branch (case 2), even gives slightly worse performance. The largest gain is when this initial shape info is added to the trunk (case 3). Here we see a decrease of 71% in the

aRRMSE and a decrease of 63.6% in the aRelErr, compared to case 1. This suggests that initial shapes should be input to the trunk, as this component of the network is responsible for learning the basis functions. Improved representation of basis functions often leads to a more accurate approximation of the solution. It is also the most logical choice because the initial shape info is more closely related to the spatial coordinates, which are also fed into the trunk, rather than to the parameters, which are input to the branch. Case 4, where the skeleton DeepONet only has the addition of the sine augmentation blocks, is the worst-performing one with $R^2 = 0.89$, aRRMSE = 0.33, and aRelErr = 0.15. From this, we conclude that the network cannot sufficiently learn the dermal displacements without knowledge of the initial shape of the wound. Additionally, sine augmentation provides an improvement in performance only when the model already performs sufficiently without it. Indeed, if we combine cases 3 and 4, we see that our proposed model overall results in the best performance.

4.3. Prediction after $t = 100$

So far, we have taken $t \in [0, 100]$ days for all predictions to save time when generating data using numerical simulations. However, we are interested to see if our final DeepONet is also capable of predicting $t \in [0, 365]$ days. Fig. 10 shows two results of evaluating the DeepONet trained on $t \in [0, 100]$ days (the original data), on an entire year. We observe that the predicted RSAW is linearly increasing after $t = 100$ days, and no asymptotic value is reached, which means the network predicts a scar that continues to grow in size. This indicates the need for adding data to the training set with $t_{end} = 365$. We experimented with different data sampling strategies. Specifically, we compare the effects of incorporating data from additional finite element simulations, to drawing more data points from existing samples. Practically, this comparison reflects the difference between including data on more burn patients versus collecting more data per patient.

We generate a new convex test set for evaluating the model's performance, consisting of 50 finite element runs with $t_{end} = 365$. Fig. 10 shows the results from three different training data sampling strategies. For each case, we initialize the network with the learned parameters from the final DeepONet trained on $t \in [0, 100]$ as a good initial guess to speed up the training. In this way, the network embeds prior knowledge instead of having to start learning from scratch.

We compare the scenario where 50 finite element simulations are added to the training set, with 30 time steps sampled for each (scenario 1), versus the scenario where 150 runs are added, with 10 time steps sampled (scenario 2). Note that both result in the same number of additional data points (30000). In both the best and worst cases, we observe that scenario 2 performs better. Sampling more data per existing finite element run does not lead to much improvement. In fact, in Fig. 10 (a), we even observe a decrease compared to selecting 10 time steps. Adding new, unique data results in a performance increase. In the best case, the predicted RSAW at $t = 365$ is now only 0.47% higher than the target. In the worst case, this is 0.6%.

The above results show that adding more runs is preferred over adding more data points within a run, possibly because adding runs increases the diversity of the training set. It exposes the network to a wider variety of features, which can enable it to generalize better. On the other hand, introducing more data per existing samples can introduce redundancy, as the network is repeatedly exposed to the same patterns. Since these patterns were already learned in the original data, additional exposure does not result in significant improvement and can lead to overfitting. This means that the network becomes too specialized in recognizing the patterns in the training data, reducing its ability to generalize to new, unseen data.
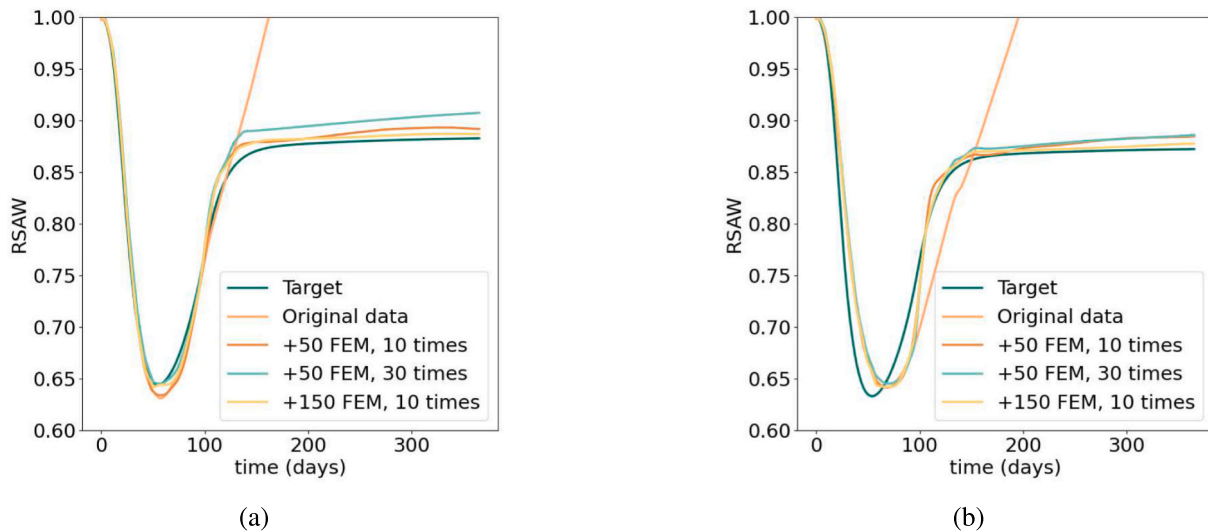
Fig. 10. Best (a) and worst (b) prediction in terms of the RSAW on a small convex test set, comparing different training data sampling strategies. The legend denotes the number of finite element simulations added to the training set and the number of time steps sampled per simulation.

4.4. Speedup

In this study, all computations were performed on the DelftBlue supercomputer, provided by the Delft High Performance Computing Centre (Delft High Performance Computing Centre (DHPC), 2024). For the finite element simulations, we used a single node equipped with an Intel Xeon Gold 6248R CPU, using one core and 2 GB of memory. Executing the finite element-based model for a one-year prediction required 4.7 min of computing time per sample on average. In contrast, using the same computational resources, the DeepONet predicted the displacement field in an average of 2.2 s per sample, resulting in a speedup of about 128 times. Furthermore, DeepONet has the advantage of being evaluated on a GPU. Using one NVIDIA A100 GPU on DelftBlue, the average prediction time per sample decreased to 1.2 s, which gives an additional speedup of almost two times. This shows that, in addition to being effective, DeepONets can serve as fast surrogates for traditional finite element methods in simulating post-burn contraction.

However, previous studies (Egberts et al., 2022; Egberts et al., 2023) reported larger speedups compared to the numerical model. In (Egberts et al., 2023), MLPs were trained for the much simpler task of predicting the RSAW and wound boundary over time, given 25 parameters from the 2D morphoelastic model. Here, a total of 21 spatial points were used to describe a fixed initial wound shape. Since this differs from our approach of predicting the entire displacement field over time for multiple wound shapes, even shapes that were not 'seen' in the training data, a direct one-to-one comparison is not possible. The CPU time needed for our DeepONet to evaluate only the nodes on the wound boundary (100 in total) for each $t \in [0, 365]$ is, on average, 0.27 s per sample. If we compare this to the execution time of the numerical model, we find a speedup of about 1044 times.

The 2.2 s reported above for the DeepONet corresponds to evaluating the DeepONet on all nodes (t, x, y) in the space-time grid. This means that, per sample evaluation, many forward passes need to be made until the network has processed all grid points in the domain, for each $t \in [0, 365]$. For example, one sample from the convex test used for one-year predictions contains an average of 180 010 points (t, x, y). Consequently, with a batch size of 50, the model has to be evaluated approximately 3600 times per sample. This is different from the previous approaches, where the MLP had to be evaluated only once to predict all quantities for each case. However, the flexibility regarding geometry realized by our DeepONet is a significant advancement, while still achieving a speedup of about 128 on CPU and 235 on GPU.

5. Discussion

A key aspect of our research was the novelty of incorporating multiple initial wound shapes. A promising direction for further research is to investigate the application of more complex and realistic initial wound geometries, where, for example, no symmetry in *x* and *y* is present. One possible approach is to include convolutional layers before the trunk network, allowing for the direct input of an image of the initial wound shape, thereby increasing the model's applicability. Additionally, an alternative to sine augmentation is to penalize discrepancies between the predicted and true displacements on the domain boundary within the loss function, adopting a PINN-like approach for softer enforcement of the boundary conditions (Raissi et al., 2019). Further integration of DeepONets with PINNs can also ensure that all predicted displacements adhere more closely to the governing PDEs, which now is not necessarily the case (Wang et al., 2021). This could solve potential discrepancies between the predictions and the underlying physical laws, leading to more reliable outcomes.

Another direction for further research is to include a broader range of patient-specific parameters as input to the branch. This approach could provide a more comprehensive and applicable model; however, it would likely introduce larger variability in the solutions since some parameters (e.g., collagen) are particularly sensitive. This would present an interesting opportunity to test whether the model can manage this increased complexity. Consequently, the architecture may need to be extended or improved.

Some approaches can be used to potentially improve performance. One of them is normalizing the magnitude of the inputs to the branch network. By ensuring that the input parameters are on a similar scale, we may help the model converge faster and prevent any one input feature from dominating the learning process. Another potential improvement for future work is increasing the variability of the training data to enhance the DeepONet's ability to generalize to more arbitrary wound shapes. Currently, the model is trained on three basic shapes, but expanding the training set to include a broader range of geometries would be a crucial step towards real data-based wounds. While it performs well on convex combinations of the three basic shapes, the shapes considered are still too limited to capture the full diversity of wound geometries. Training on a wider and more diverse set of shapes would improve generalization, enabling the model to handle more arbitrary and irregular wound structures effectively. However, our results show a promising proof of concept in the sense that a set of basis wound

geometries can be used so that a convex combination of these basis geometries can be modelled reliably. In other words, wound geometries that have not been 'seen' in the training set can still be simulated with high fidelity. We expect that by adding more basis geometries, even more general and complicated wound geometries can be simulated.

6. Conclusions

We trained a DeepONet on three different initial wound shapes to accurately predict the dermal displacement across the entire domain over time. The DeepONet consists of a branch and trunk network. The branch network takes as input five parameters from the morphoelastic model describing burn injuries: the cell diffusion constant, the chemotactic constant, the signaling diffusion constant, the cell differentiation rate, and the half-maximum cell division enhancement rate. The trunk takes as input a coordinate (t, x, y) where we want to evaluate the solution, along with information about the initial wound shape and size given by the quadruple $(y_{cut}, x_m, y_m, x_{cut})$. The final DeepONet output is the result of a dot product of the branch and trunk network outputs, such that the five input parameters are mapped to the displacement fields for 100 post-burn days.

We evaluated the performance of the trained DeepONet on a test set including finite element simulations based on convex combinations of the three basic wound shapes. We found that the predictions closely matched targets, demonstrating good generalization. The scatter plots in Fig. 8 confirm this, showing a strong concentration around $y = x$ with minimal outliers. The model slightly underperformed on mid-range y-displacements but excelled in predicting both small and large displacements. The error analysis in Section 4 showed a gradual increase in absolute error over time, peaking around day 90. However, the mean plus standard deviation of the error never exceeds 3%, which corresponds to deviations on the order of tenths of millimeters. From a practical medical perspective, these are very accurate predictions.

We compared our various DeepONet architectures on the convex test set, cf. Fig. 3. The final model, incorporating initial wound shape info and sine augmentation, achieved the best performance with an R^2 score of 0.99, an aRRMSE of 0.08, and an aRelErr of 0.03. This confirms the model's strong generalization abilities. The results support that initial shape info should be input to the trunk, and sine augmentation enhances performance when the model is already well-tuned.

Finally, we extended the DeepONet predictions to one year rather than 100 days. We expanded the training set with 50 new finite element simulations, where $t_{end} = 365$. We compared the RSAW curves for two scenarios: one where 50 finite element simulations were added to the training set, with 30 time steps sampled in $t \in (100, 365]$, and another where 150 runs were added, with 10 time steps each. We found that adding new, unique data (the first scenario) results in the greatest improvement. Practically, these findings suggest that including more patients with less data per patient is preferable to having fewer patients with more data per patient. This is likely because introducing new information provides a more varied learning experience, whereas

additional data points from existing samples tend to add redundancy, offering less benefit.

We conclude that we have successfully trained a DeepONet as a finite element surrogate, capable of accurately predicting the post-burn dermal displacement field. The training encompassed multiple initial wound shapes, with the network demonstrating strong generalization on convex combinations of these basic shapes. Furthermore, by adding a limited number of samples from one-year finite element simulations into the training set, the network achieved reasonable predictions of the RSAW over the entire year. We anticipate that including more data with $t_{end} = 365$ will further enhance prediction accuracy.

As a final conclusion, we experimentally found that the speedup of a simulation run using the DeepONet is about a factor 128 relative to the original finite element simulations (235 when evaluated on a GPU). While this speedup is lower than that achieved in (Egberts et al., 2022; Egberts et al., 2023) for MLP models on a fixed geometry, the proposed DeepONet allows for geometrical flexibility that was not possible with the MLPs employed in previous works.

Datasets and code

All relevant code is available at github.com/Selma24/DeepONet-contraction. All relevant data is available on the 4TU.Centre for Research Data at doi.org/10.4121/69d1aefc-a01d-4280-8b32-5c8420d9a2a3.

CRediT authorship contribution statement

Selma Husanovic: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Ginger Egberts:** Writing – review & editing, Writing – original draft, Supervision, Software, Resources, Project administration. **Alexander Heinlein:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Fred Vermolen:** Writing – review & editing, Writing – original draft, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Appendix

Table 3 lists the fixed parameter values used for the finite element simulations that were not taken as input to the DeepONet's branch network.

Table 3

Overview of the fixed parameter values used for the numerical simulations. 'NC' indicates that the parameter value is a consequence of the chosen values for other parameters.

Symbol	Value	Dimension	Reference
k_c	$4 \cdot 10^{-13}$	g/(cells day)	(Olsen et al., 1995)
r_F	$9.24 \cdot 10^{-1}$	cm ^{3q}/(cells ^{q} day)	(Alberts et al., 1989; Gosh et al., 2007)
$r_F^{\max}$	2	–	(Strutz et al., 2001)
k_ρ	$7.6 \cdot 10^{-8}$	g/(cells day)	[NC]
$k_\rho^{\max}$	10	–	(Olsen et al., 1995)
a_c^{II}	10^{-8}	g/cm ^{3}	(Olsen et al., 1995)
a_c^{III}	$2 \cdot 10^8$	cm ^{3}/g	(Overall et al., 1991)
a_c^{IV}	10^{-9}	g/cm ^{3}	(Roberts et al., 1986)
η^I	2	–	(Rudolph and Vande Berg, 1991)
η^{II}	$5 \cdot 10^{-1}$	–	(Koppenol and Vermolen, 2017)
κ_F	10^{-6}	cm ^{3}/cells	(Vande Berg et al., 1989)
q	$-4.151 \cdot 10^{-1}$	–	[NC]
δ_N	$2 \cdot 10^{-2}$	/day	(Olsen et al., 1995)
δ_M	$6 \cdot 10^{-2}$	/day	(Koppenol et al., 2017)
δ_c	$5 \cdot 10^{-4}$	cm ^{6}/(cells g day)	(Olsen et al., 1995)
δ_ρ	$6 \cdot 10^{-6}$	cm ^{6}/(cells g day)	(Koppenol et al., 2017)
$\overline{N}$	10^4	cells/cm ^{3}	(Olsen et al., 1995)
$\overline{M}$	0	cells/cm ^{3}	(Olsen et al., 1995)
$\overline{c}$	0	g/cm ^{3}	(Koppenol et al., 2017)
$\overline{\rho}$	$1.125 \cdot 10^{-1}$	g/cm ^{3}	(Olsen et al., 1995)
ρ_t	1.09	g/cm ^{3}	(ICRP, 2010)
μ_1	10^2	(N day)/cm ^{2}	(Koppenol and Vermolen, 2017)
μ_2	10^2	(N day)/cm ^{2}	(Koppenol and Vermolen, 2017)
E	32	N/((g cm)$^{0.5}$)	(Egberts et al., 2021)
ξ	$5 \cdot 10^{-2}$	(N g)/(cells cm ^{2})	(Maskarinec et al., 2009; Wrobel et al., 2002)
R	$9.95 \cdot 10^{-1}$	g/cm ^{3}	(Koppenol and Vermolen, 2017)
ζ	$4 \cdot 10^2$	cm ^{6}/(cells g day)	(Koppenol and Vermolen, 2017)
ν	$4.9 \cdot 10^{-1}$	–	(Egberts et al., 2021)
$\tilde{N}$	$2 \cdot 10^3$	cells/cm ^{3}	[NC]
$\tilde{c}$	10^{-8}	g/cm ^{3}	(Egberts et al., 2021)
$\tilde{\rho}$	$1.13 \cdot 10^{-2}$	g/cm ^{3}	[NC]

References

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J., 1989. The Molecular Biology of the Cell, Second ed. Garland Publishing.

Barocas, V., Tranquillo, R., 1997. An anisotropic biphasic theory of tissue-equivalent mechanics: the interplay among cell traction, fibrillar network deformation, fibril alignment, and cell contact guidance. J. Biomech. Eng. 119, 137–145. https://doi.org/10.1115/1.2796072.

Chen, T., Chen, H., 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. IEEE Trans. Neural Netw. 6, 911–917.

Dallon, J., Sherrat, J., Maini, P., 1999. Mathematical modelling of extracellular matrix dynamics using discrete cells: fiber orientation and tissue regeneration. J. Theor. Biol. 199, 449–471. https://doi.org/10.1006/jtbi.1999.0971.

Delft High Performance Computing Centre (DHPC), 2024. DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

Desmoulière, A., Geinoz, A., Gabbiani, F., Gabbiani, G., 1993. Transforming growth factor-β1 induces alpha-smooth muscle actin expression in granulation tissue myofibroblasts and in quiescent and growing cultured fibroblasts. J. Cell Biol. 122, 103–111. https://doi.org/10.1083/jcb.122.1.103.

Despotovic, M., Nedic, V., Despotovic, D., Cvetanovic, S., 2016. Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation. Renew. Sust. Energ. Rev. 56, 246–260.

Diab, W., Al-Kobaisi, M., . U-DeepONet: U-Net Enhanced Deep Operator Network for Geologic Carbon Sequestration. http://arxiv.org/abs/2311.15288. https://doi.org/10.48550/arXiv.2311.15288. arXiv:2311.15288 [physics].

Egberts, G., Smits, D., Vermolen, F., van Zuijlen, P., 2020. Some mathematical properties of morphoelasticity. In: Lecture Notes in Computational Science and Engineering. Springer International Publishing, pp. 1119–1127. https://doi.org/10.1007/978-3-030-55874-1_111.

Egberts, G., Vermolen, F., van Zuijlen, P., 2021. Sensitivity and feasibility of a one-dimen-sional morphoelastic model for post-burn contraction. Biomech. Model. Mechanobiol. 20, 2147–2167. https://doi.org/10.1007/s10237-021-01499-5.

Egberts, G., Schaaphok, M., Vermolen, F., van Zuijlen, P., 2022. A Bayesian finite-element trained machine learning approach for predicting post-burn contraction. Neural Comput. & Applic. 34, 8635–8642. https://doi.org/10.1007/s00521-021-06772-3.

Egberts, G., Vermolen, F., van Zuijlen, P., 2023. High-speed predictions of post-burn contraction using a neural network trained on 2d-finite element simulations. Front. Appl. Math. Stat. 9. https://doi.org/10.3389/fams.2023.1098242.

Goel, A., Shrivastava, P., 2010. Post-burn scars and scar contractures. Indian J. Plast. Surg. 43, 63. https://doi.org/10.4103/0970-0358.70724.

Goodfellow, I., Bengio, Y., Courville, A.,. Deep Learning. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA https://mathscinet.ams.org/mathscinet-getitem?mr=3617773.

Gosh, K., Pan, Z., Guan, E., Ge, S., Lio, Y., Nakamura, T., Ren, Z., Rafailovich, M., Clark, R., 2007. Cell adaptation to a physiologically relevant ECM mimic with different viscoelastic properties. Biomaterials 28, 671–679. https://doi.org/10.1016/j.biomaterials.2006.09.038.

Griese, F., Hoppe, F., Rüttgers, A., Knechtges, P., . FEM-Based Neural Networks for Solving Incompressible Fluid Flows and Related Inverse Problems. http://arxiv.org/abs/2409.04067. https://doi.org/10.48550/arXiv.2409.04067. arXiv:2409.04067 [physics].

Hall, C.L., 2008. Modelling of some Biological Materials Using Continuum Mechanics. Ph.D. thesis. Queensland University of Technology.

Heinlein, A., Howard, A.A., Beecroft, D., Stinis, P., . Multifidelity Domain Decomposition-Based Physics-Informed Neural Networks for Time-Dependent Problems. http://arxiv.org/abs/2401.07888. https://doi.org/10.48550/arXiv.2401.07888. arXiv:2401.07888 [cs, math].

Hillen, T., Painter, K.J., 2009. A user's guide to pde models for chemotaxis. J. Math. Biol. 58, 183–217.

Howard, A.A., Murphy, S.H., Ahmed, S.E., Stinis, P., . Stacked Networks Improve Physics-Informed Training: Applications to Neural Networks and Deep Operator Networks. http://arxiv.org/abs/2311.06483. https://doi.org/10.48550/arXiv.2311.06483. arXiv:2311.06483 [cs, math].

Husanovic, S., 2024. The Application of Neural Operators to Predict Skin Evolution after Burn Trauma. Unpublished thesis.

ICRP, 2010. ICRP Publication 110. SAGE Publications Ltd, p. 51.

Kelley, H.J.,. Gradient Theory of Optimal Flight Paths 30, 947–954. doi: https://doi.org/10.2514/8.5282, American Institute of Aeronautics and Astronautics.

Kingma, D.P., Ba, J., (2014) Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980, doi:10.48550/arXiv.1412.6980, 1412.6980 [cs].

Kopaničáková, A., Karniadakis, G.E.. DeepOnet Based Preconditioning Strategies for Solving Parametric Linear Systems of Equations. http://arxiv.org/abs/2401.02016 arXiv:2401.02016 [cs, math].

Koppenol, D., 2017. Biomedical Implications from Mathematical Models for the Simulation of Dermal Wound Healing. Ph.D. thesis. Delft University of Technology.

Koppenol, D., Vermolen, F., 2017. Biomedical implications from a morphoelastic continuum model for the simulation of contracture formation in skin grafts that cover excised burns. Biomech. Model. Mechanobiol. 16, 1187–1206. https://doi. org/10.1007/s10237-017-0881-y.

Koppenol, D.C., Vermolen, F.J., Niessen, F.B., van Zuijlen, P.P., Vuik, K., 2017. A mathematical model for the simulation of the formation and the subsequent regression of hypertrophic scar tissue after dermal wounding. Biomech. Model. Mechanobiol. 16, 15–32.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A.,. Neural Operator: Learning Maps Between Function Spaces. URL: http://arxiv.org/abs/2108.08481, doi:10.48550/arXiv.2108.08481, arXiv:2108.08481 [cs, math].

Lagaris, I., Likas, A., Fotiadis, D., 1998. Artificial neural networks for solving ordinary and partial differential equations. IEEE Trans. Neural Netw. 9, 987–1000. https:// doi.org/10.1109/72.712178 conference Name: IEEE Transactions on Neural Networks.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.,. Fourier Neural Operator for Parametric Partial Differential Equations http://arxiv.org/abs/2010.08895, doi: 10.48550/arXiv.2010.08895, arXiv:2010.08895 [cs, math].

Lin, G., Moya, C., Zhang, Z., 2023. Learning the dynamical response of nonlinear non-autonomous dynamical systems with deep operator neural networks. Eng. Appl. Artif. Intell. 125, 106689.

Liu, D.C., Nocedal, J.. On the Limited Memory BFGS Method for Large Scale Optimization Vol. 45, 503–528. https://doi.org/10.1007/BF01589116.

Lu, L., Jin, P., Karniadakis, G.E. (2021) DeepONet: Learning Nonlinear Operators for Identifying Differential Equations Based on the Universal Approximation Theorem of Operators Vol. 3, 218–229. http://arxiv.org/abs/1910.03193, doi:htt ps://doi.org/10.1038/s42256-021-00302-5, arXiv:1910.03193 [cs, stat].

Maskarinec, S., Franck, C., Tirell, D., Ravichandran, G., 2009. Quantifying cellular traction forces in three dimensions. Proc. Natl. Acad. Sci. 106, 22108–22113. https://doi.org/10.1073/pnas.0904565106.

Meethal, R.E., Kodakkal, A., Khalil, M., Ghantasala, A., Obst, B., Bletzinger, K.U., Wüchner, R., 2023. Finite element method-enhanced neural network for forward and inverse problems. Adv. Model. Simul. Eng. Sci. 10, 6. https://doi.org/10.1186/ s40323-023-00243-1.

Menon, S., Hall, C., McCue, S., McElwain, D., 2017. A model for one-dimensional morphoelasticity and its application to fibroblast-populated collagen lattices. Biomech. Model. Mechanobiol. 16, 1743–1763. https://doi.org/10.1007/s10237- 017-0917-3.

Möller, M., Kuzmin, D., Kourounis, D., 2008. Implicit FEM-FCT algorithms and discrete Newton methods for transient convection problems. Int. J. Numer. Methods Fluids 57, 761–792. https://doi.org/10.1002/fld.1654.

Möller, M., Toshniwal, D., van Ruiten, F., 2021. Physics-Informed Machine Learning Embedded into Isogeometric Analysis. Platform Wiskunde, Amsterdam, pp. 57–59.

Olsen, L., Sherratt, J., Maini, P., 1995. A mechanochemical model for adult dermal wound contraction and the permanence of the contracted tissue displacement profile. J. Theor. Biol. 177, 113–128. https://doi.org/10.1006/jtbi.1995.0230.

Overall, C., Wrana, J., Sodek, J., 1991. Transcriptional and post-transcriptional regulation of 72-kda gelatinase/ type IV collagenase by transforming growth factor-beta in human fibroblasts. J. Biol. Chem. 266, 14061–14071.

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. https://mathscinet.ams.org/mathscinet-getitem?mr=3881695. htt ps://doi.org/10.1016/j.jcp.2018.10.045.

Raonić, B., Molinaro, R., De Ryck, T., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., de Bézenac, E.,. Convolutional Neural Operators for Robust and Accurate Learning of PDEs. http://arxiv.org/abs/2302.01178, doi: 10.48550/arXiv.2302.01178, arXiv:2302.01178 [cs].

Roberts, A., Sporn, M., Assoian, R., Smith, J., Roche, N., Wakefield, L., Heine, U., Liotta, L., Falanga, V., Kehrl, J., 1986. Transforming growth factor type beta: rapid induction of fibrosis and angiogenesis in vivo and stimulation of collagen formation in vitro. Proc. Natl. Acad. Sci. 83, 4167–4171. https://doi.org/10.1073/ pnas.83.12.4167.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Springer International Publishing, Cham, pp. 234–241.

Rudolph, R., Vande Berg, J., 1991. The myofibroblast in Dupuytren's contracture. Journal of. Hand Clin. 7, 683–692 (Discussion 693-4).

Stéphanou, A., Volpert, V., 2015. Hybrid modelling in biology: a classification review. Math. Model. Nat. Phenom. 11, 37–48. https://doi.org/10.1051/mmnp/201611103.

Strutz, F., Zeisberg, M., Renziehausen, A., Raschke, B., Becker, V., Van Kooten, C., Müller, G., 2001. TGF-β 1 induces proliferation in human renal fibroblasts via induction of basic fibroblast growth factor (FGF-2). Kidney Int. 59, 579–592. https:// doi.org/10.1046/j.1523-1755.2001.059002579.x.

Sun, Y., Moya, C., Lin, G., Yue, M.,. DeepGraphONet: A Deep Graph Operator Network to Learn and Zero-Shot Transfer the Dynamic Response of Networked Systems. http://arxiv.org/abs/2209.10622. https://doi.org/10.48550/ arXiv.2209.10622. arXiv:2209.10622 [cs].

Tomasek, J., Gabbiani, G., Hinz, B., Chaponnier, C., Brown, R., 2002. Myofibroblasts and mechano-regulation of connective tissue remodelling. Nat. Rev. Mol. Cell Biol. 3, 349–363. https://doi.org/10.1038/nrm809.

Vande Berg, J., Rudolph, R., Poolman, W., Disharoon, D., 1989. Comparative growth dynamics and actin concentration between cultured human myofibroblasts from granulating wounds and dermal fibroblasts from normal skin. Lab. Investig. 61, 532–538.

Wang, Y., Beekman, J., Hew, J., Jackson, S., Issler-Fisher, A., Parungao, R., Lajevardi, S., Li, Z., Maitz, P., 2018. Burn injury: challenges and advances in burn wound healing, infection, pain and scarring. Adv. Drug Deliv. Rev. 123, 3–17. https://doi.org/ 10.1016/j.addr.2017.09.018.

Wang, Y., Cheung, S.W., Chung, E.T., Efendiev, Y., Wang, M., 2020. Deep multiscale model learning. J. Comput. Phys. 406, 109071. https://www.sciencedirect. com/science/article/pii/S0021999119307764. https://doi.org/10.1016/j. jcp.2019.109071.

Wang, S., Wang, H., Perdikaris, P., 2021. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. Sci. Adv. 7, eabi8605.

WHO, 2018. World Health Organisation, Fact Sheet, Burns, 06-03-2018. https://www. who.int/en/news-room/fact-sheets/detail/burns. Accessed on 04-12- 2019.

Wrobel, L., Fray, T., Molloy, J., Adams, J., Armitage, M., Sparrow, J., 2002. Contractility of single human dermal myofibroblasts and fibroblasts. Cell Motil. Cytoskeleton 52, 82–90. https://doi.org/10.1002/cm.10034.

Xiao, X., Feng, X., Yuan, J., 2018. The lumped mass finite element method for surface parabolic problems: error estimates and maximum principle. Comput. Math. Appl. 76 (3), 488–507.

Zhang, Z., Wing Tat, L., Schaeffer, H., 2023. Belnet: basis enhanced learning, a mesh-free neural operator. Proc. R. Soc. A 479, 20230043.

Zhang, Z., Moya, C., Lu, L., Lin, G., Schaeffer, H., 2024. D2no: efficient handling of heterogeneous input function spaces with distributed deep neural operators. Comput. Methods Appl. Mech. Eng. 428, 117084.

Zhang, Z., Leung, W.T., Schaeffer, H., 2025. A discretization-invariant extension and analysis of some deep operator networks. J. Comput. Appl. Math. 456, 116226.