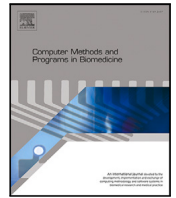




Contents lists available at ScienceDirect

# Computer Methods and Programs in Biomedicine

journal homepage: <https://www.sciencedirect.com/journal/computer-methods-and-programs-in-biomedicine>



## Deep vectorised operators for pulsatile hemodynamics estimation in coronary arteries from a steady-state prior

Julian Suk<sup>a</sup>,<sup>\*</sup>, Guido Nannini<sup>b</sup>, Patryk Rygiel<sup>a</sup>, Christoph Brune<sup>a</sup>, Gianluca Pontone<sup>c,d</sup>,  
Alberto Redaelli<sup>b</sup>, Jelmer M. Wolterink<sup>a</sup>

<sup>a</sup> Department of Applied Mathematics and Technical Medical Center, University of Twente, Enschede, The Netherlands

<sup>b</sup> Department of Electronics Information and Bioengineering, Politecnico di Milano, Milan, Italy

<sup>c</sup> Department of Perioperative Cardiology and Cardiovascular Imaging, Centro Cardiologico Monzino IRCCS, Milan, Italy

<sup>d</sup> Department of Biomedical, Surgical and Dental Sciences, University of Milan, Milan, Italy

### ARTICLE INFO

#### Keywords:

Coronary hemodynamics  
Computational fluid dynamics  
Machine learning  
Coronary simulation runtime

### ABSTRACT

**Background and objective:** Cardiovascular hemodynamic fields provide valuable medical decision markers for coronary artery disease. Computational fluid dynamics (CFD) is the gold standard for accurate, non-invasive evaluation of these quantities *in silico*. In this work, we propose a time-efficient surrogate model, powered by machine learning, for the estimation of pulsatile hemodynamics based on steady-state priors.

**Methods:** We introduce deep vectorised operators, a modelling framework for discretisation-independent learning on infinite-dimensional function spaces. The underlying neural architecture is a neural field conditioned on hemodynamic boundary conditions. Importantly, we show how relaxing the requirement of point-wise action to permutation-equivariance leads to a family of models that can be parametrised by message passing and self-attention layers. We evaluate our approach on a dataset of 74 stenotic coronary arteries extracted from coronary computed tomography angiography (CCTA) with patient-specific pulsatile CFD simulations as ground truth.

**Results:** We show that our model produces accurate estimates of the pulsatile velocity and pressure (approximation disparity  $0.368 \pm 0.079$ ) while being agnostic ( $p < 0.05$  in a one-way ANOVA test) to re-sampling of the source domain, i.e. discretisation-independent.

**Conclusion:** This shows that deep vectorised operators are a powerful modelling tool for cardiovascular hemodynamics estimation in coronary arteries and beyond.

### 1. Introduction

Coronary artery disease (CAD) consists in the accumulation of plaque in the artery wall which results in narrowing of the blood vessel. Stenoses then impair the flow of oxygenated blood through the coronary arteries to the heart muscle, which can lead to myocardial infarction [1]. Visualisation and quantification of patient-specific blood flow can be valuable for diagnosis [2], prognosis [3] and treatment [4] of cardiovascular disease. Quantities derived from the blood flow and pressure, e.g., wall shear stress, oscillatory shear index and fractional flow reserve (trans-stenotic pressure ratio, measured during maximal hyperemia), have been identified as useful decision markers and risk factors in CAD. For instance, wall shear stress can act as a predictor for culprit lesions in cases in which fractional flow reserve is above threshold ( $>0.8$ ) but the lesion is ischemic [5]. Wall shear stress and other biomarkers derived from blood flow are indicated as *possible*

*alternatives* to detect ischemic lesions. Computing these biomarkers requires access to pulsatile velocity and pressure fields within the vessel.

Clinical hemodynamic biomarkers can be non-invasively quantified *in-vivo* with imaging techniques such as Doppler ultrasound, particle image velocimetry or 4D flow magnetic resonance imaging (MRI). However, the small size of the coronary arteries limits the extraction of hemodynamic metrics that can effectively distinguish pathological conditions. Alternatively, these metrics can be obtained *in-silico* via computational fluid dynamics (CFD) based on 3D patient models reconstructed from computed tomography (CT) [6]. Some of the challenges related to clinical adoption of CFD are its long runtimes, high computational demand and dependence on expert knowledge throughout the process. Furthermore, sensitivity to modelling choices, such as *discretisation* of time and space, as well as boundary conditions, make

\* Corresponding author.

E-mail address: [j.m.suk@utwente.nl](mailto:j.m.suk@utwente.nl) (J. Suk).

<https://doi.org/10.1016/j.cmpb.2025.108958>

Received 16 October 2024; Received in revised form 5 June 2025; Accepted 8 July 2025

Available online 28 July 2025

0169-2607/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

it difficult to decide upon systematic protocols that are consistent and repeatable across hospitals. Indeed, it has been shown that there is inter-operator variability between CFD simulations based on identical medical images [7]. While CFD is gradually making its way into the clinic, pioneered in part by commercial service providers like HeartFlow [8], the above challenges make such services expensive which may hinder widespread adoption.

In this work, we address these challenges with an in-silico surrogate model for estimation of coronary hemodynamics, based on machine learning. We train the model on a dataset of pulsatile CFD simulations, i.e. time-varying velocity and pressure fields, in right and left coronary branches of 74 patients scheduled for clinically indicated invasive coronary angiography for suspected CAD. Additionally, we create a steady-state simulation for each patient, representing the pulsatile CFD solution at the first instant of the cardiac cycle (i.e., at time  $t = t_0 + dt$  where  $t_0$  is the end diastolic time) and use them to cast the learning objective as operator learning: we let our neural networks learn a mapping from the steady-state to the corresponding pulsatile hemodynamics. The steady-state CFD solution provides a powerful prior that enables us to faithfully predict the pulsatile hemodynamics even with limited (yet diverse) data.

We name our approach *deep vectorised operators* and lay out a modelling framework in which the nonlinear operator is learned by a vectorised conditional neural field. Relaxing the requirement of point-wise action of neural fields to permutation equivariance allows us to parametrise our models as PointNet++ [9] and transformers [10] which have recently proven capable of learning hemodynamics in large-scale, anatomical meshes [11,12]. To further facilitate learning from limited amounts of data, we incorporate group equivariance in our models, which can be advantageous for extracting information from small hemodynamic datasets [13]. As in CFD [14], our method is informed by patient-specific boundary conditions. Once trained, our models can quickly produce pressure and velocity field estimates for vascular anatomy on consumer hardware based on boundary conditions for a new patient. We empirically show that our model is robust w.r.t. discretisation of the flow domain.

Our contributions are as follows: (1) we explore learning pulsatile hemodynamics from steady-state priors with a nonlinear operator, (2) we provide a unified modelling framework for neural hemodynamic field estimation with state-of-the-art neural architectures and (3) we evaluate our approach on a relevant, real-life dataset of coronary arteries with patient-specific CFD simulations.

### 1.1. Related works

Estimation of hemodynamics with machine learning methods has been an active area of research. Typically, these works fall into one of two categories: (1) transductive instance optimisation with physics-based regularisation (solving partial differential equations) and (2) inductive, generalising feed-forward methods that learn to infer hemodynamics based on patient anatomy and boundary conditions. The former category requires long training times for each new subject, often in the same order of magnitude as CFD. These methods can be useful in scenarios where hemodynamic parameters are only partially known [15–17]. In contrast, methods in the latter category are designed as fast, compute-efficient CFD surrogates.

In this study, we concentrate on the second approach: leveraging neural networks to model the relationship between vascular geometry, boundary conditions and hemodynamic patterns. These neural networks process 3D vessel structures to predict blood-flow-related parameters on the vessel surface or within the interior. In addition to research aimed at predicting 1D quantities along the vessel tree [18] and surface characteristics such as wall shear stress [19,20], pressure drop [21] as well as endothelial cell activation potential [11], several studies have focused on predicting volumetric vector fields.

For example, Liang et al. [22] and Wang et al. [23] trained fully-connected neural networks that work on 3D point-cloud representations of the carotid artery and thoracic aorta, respectively, to predict pressure and velocity fields. Maul et al. [24] developed an octree-based neural network, paired with trilinear interpolation to learn parametric, pulsatile flow independently of spatial discretisation and applied it to synthetic vascular trees. Li et al. [25] employed a point-cloud-based architecture to estimate pressure and velocity fields in coronary arteries and synthetic cerebral aneurysms. Zhang et al. [26] used a combination of PointNet++ [9] and a physics-informed neural network (PINN) based on the Navier–Stokes equations to estimate velocity fields in 3D models of the abdominal aorta. In prior work [13] we used a multiscale, E(3)-steerable graph neural network to estimate velocity fields in synthetic coronary arteries.

Wessels et al. [27] recently proposed equivariant neural fields by cross-conditioning on latent geometric attributes using transformers. This differs from our approach in that the field domain and geometric object are decoupled while in our case they coincide.

Operator learning approaches, such as DeepONet [28] and (Fourier) neural operators [29,30] have recently gained traction in the machine learning community but there are few applications to cardiovascular hemodynamics estimation. Alkin et al. [31] recently proposed a transformer-based encoder–decoder model to learn neural operators for general physical systems.

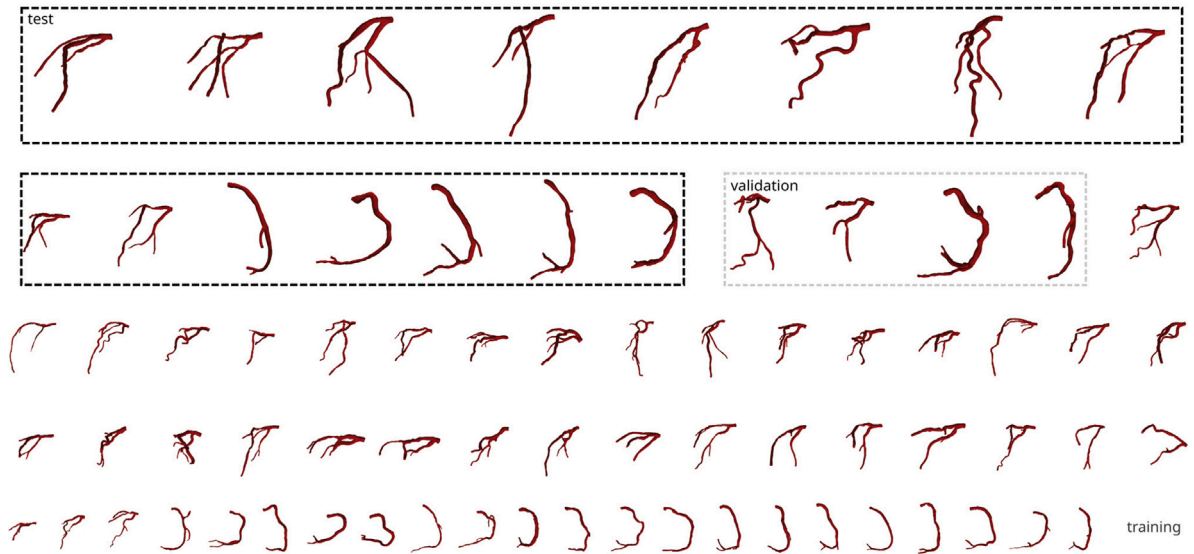
## 2. Materials and methods

### 2.1. Coronary artery dataset

Patient-specific CFD simulations of coronary flow from a previous study [6] served as the dataset. A brief overview of the data generation process is provided in the following. Coronary computed tomography angiography (CCTA) scans of patients scheduled for clinically indicated invasive coronary angiography due to suspected CAD were retrospectively collected from Centro Cardiologico Monzino (Milan, Italy). Overall, 74 stenotic vessel geometries (shown in Fig. 1), from both the left (48) and right (26) coronary arteries, were reconstructed from CCTA scans using 3DSlicer [32]. The segmentation included the artery branches downstream of the ostium, and was terminated when the vessel diameter fell below 1.5 mm, in accordance with established guidelines [33]. The aortic root was excluded. The study was performed in accordance with recommendations of the local Ethics Committee, with written informed consent from all subjects, in accordance with the Declaration of Helsinki.

The volume of the reconstructed geometries was discretised – after mesh sensitivity analysis (details in supplementary material) – into tetrahedral elements with a characteristic size of 0.25 mm using the TetGen algorithm embedded within SimVascular [34]. This led to 1,801,029 tetrahedral elements and 314,525 vertices on average. For each mesh, three cardiac cycles were simulated by numerically solving the incompressible Navier–Stokes equations, with time-varying boundary conditions, in SimVascular using svSolver while neglecting any effect related to arterial wall motion and deformation. A time discretisation of  $dt = 0.001$  s was adopted in the CFD, results were exported every 0.025 s, which corresponds to the largest time step value that ensured the numerical convergence of the simulation (based on separately run sample-tests). Results were retained solely from the last cycle, spanning 35 time steps on average.

To set up the simulation, the average hyperemic flow rate,  $\bar{Q}_{hyp}$ , was computed using clinical data. Briefly, the myocardial mass  $M_{myo}$ , heart rate  $H$  and systolic pressure ( $P_{sys}$ ) were used to compute the resting flow rate using a formula by Sharma et al. [35]:  $Q_{rest} = 0.14 \cdot (7E - 4 \cdot H \cdot P_{sys} - 0.4) \cdot M_{myo}$ . The hyperemic flow was set to 3.5-fold the resting one and used to scale a flow rate waveform obtained from the literature [36], which was applied as the inlet boundary condition (BC)  $Q_{hyp}(t)$  in the simulation. A five-element Windkessel model was



**Fig. 1.** Coronary artery dataset consisting of 74 3D models of stenotic left and right coronary arteries with different numbers of bifurcations and outlets. Patients in this cohort were scheduled for invasive coronary angiography due to suspected CAD. We partitioned the dataset into training, validation and test split while ensuring equal ratios of left and right coronary arteries between training and test split.

coupled to each outlet as the BC, with its parameters tuned to match the patient’s mean aortic pressure, for resistance, and pulse pressure, for compliance (for further details on the CFD settings, see [6]). Pulsatile simulations took between 12 to 24 h per case on a 40 cores Intel Xeon CPU X5670 machine with 64.4 GB RAM.

Additionally, a steady-state CFD simulation was performed for each model using the inflow value corresponding to the flowrate at the first pulsatile-CFD solution output:  $Q_{\text{hyp}}(t = 0.025)$ . We chose this initial time point for the steady-state simulation to minimise inertial effects, which can influence the pressure field in pulsatile CFD simulations. Simulations took between 5 to 30 min depending on the flow rate where computational cost scaled roughly with the Reynolds number. In the context of our machine learning model, the steady-state solution functions as cheap information about the characteristic flow in the respective artery. It can be thought of as the fluid-dynamic system response to an “impulse excitation”.

## 2.2. Background

In the following, we will discuss the relevant theoretical background to put our approach into context. However, our method can be understood independently of this section.

In this work we consider the problem of learning time-dependent,  $c$ -dimensional vector fields ( $c \in \mathbb{N}$ ) from data

$$y: \begin{cases} T \times \Omega \rightarrow \mathbb{R}^c \\ t, x \mapsto y(t, x) \end{cases} \quad T \subset \mathbb{R}, \quad \Omega \subset \mathbb{R}^3,$$

which are defined on a time interval  $T$  and spatial domain  $\Omega$ . In the following we propose a family of models that can efficiently represent specific instances of such vector fields while enabling zero-shot *generalisation* to instances beyond the training data. In contrast to common applications in medical imaging, we do not assume the data to have grid structure that we can exploit, e.g., via convolutional neural networks (CNN). Rather, our model learns a map between infinite-dimensional function spaces.

### 2.2.1. Neural fields

A neural field is a parametrised map  $f_\theta: \mathbb{R}^{c_{\text{in}}} \rightarrow \mathbb{R}^{c_{\text{out}}}$  between vector spaces where  $f_\theta$  is a neural network with parameters  $\theta$ , e.g., a multilayer perceptron (MLP). The domain  $\mathbb{R}^{c_{\text{in}}}$  is commonly required to be spacetime [37]. The neural network  $f_\theta$  is trained under observations

$\{(x^i \in \mathbb{R}^{c_{\text{in}}}, y^i \in \mathbb{R}^{c_{\text{out}}})\}_{i \in [1, n]}$  of input–output pairs.<sup>1</sup> Since neural networks are commonly continuous or continuously differentiable, neural fields can be smooth representations of scalar and vector fields in a variety of applications, such as neural signed distance functions [38] and physics-informed neural networks [39].

*Conditional neural fields.* Neural fields can be conditioned on a collection of parameters  $\xi \in \Xi$ , e.g., by concatenating ( $\|$ ) them with the input observations and feeding them to the neural network

$$f_\theta: \begin{cases} \mathbb{R}^{c_{\text{in}}} \times \Xi \rightarrow \mathbb{R}^{c_{\text{out}}} \\ x, \xi \mapsto f_\theta(x \| \xi). \end{cases}$$

This enables, e.g., representing multiple vector fields with the same domain and co-domain in a single neural field by assigning each vector field a latent code, as proposed by DeepSDF [40], or representing *parametric* solutions to partial differential equations. In this case, observations depend on the parameters  $\xi$  under which they were made.

*Vectorised conditional neural fields.* When training conditional neural fields, input observations  $x^i$  are commonly vectorised into an  $(n \times c_{\text{in}})$ -dimensional tensor while  $f_\theta$  (conditioned on  $\xi$ ) broadcasts over the batch dimension. Accordingly,  $f_\theta$  must act point-wise, i.e., row-wise on the input tensor. However, if we always feed observations to the neural field as complete sets, we can relax this requirement and choose  $f_\theta$  – instead of point-wise – as point-wise *permutation-equivariant*. Such models allow introducing inter-observation context and thus implicit conditioning on the complete set of observations. Based on the same idea, Hagnberger et al. [41] proposed this family of models under the name “vectorised conditional neural fields”. This inter-observation context conditions the neural field on characteristics of the input (e.g. topology or global statistics) and supports interactions in input spacetime. Precisely this cross-conditioning gives rise to the expressive power of set-based neural networks like PointNet++ and transformers. As drawback of adding context, sets must always be processed in their entirety. In contrast, point-wise independent processing natively enables sub-dividing the sets into chunks which is useful in scenarios where memory is a bottleneck. We identify permutation equivariance as the key requirement with which set-based neural networks can be cast as *vectorised* neural fields.

<sup>1</sup>  $c_{\text{in}}$  and  $c_{\text{out}}$  are the dimensions of domain and co-domain (respectively) and  $n$  is the total number of observations.

### 2.2.2. Permutation-equivariant neural networks

Permutation-equivariant functions satisfy the condition

$$(f_{\theta} \circ P)(\cdot) \equiv (P \circ f_{\theta})(\cdot) \quad \text{for all } P \in S_n,$$

where  $\circ$  denotes composition and  $S_n$  is the symmetric group of permutations on  $n$  set elements. In other words, the relation between input and output elements cannot depend on their specific ordering. Note that point-wise functions are trivially permutation-equivariant. Among the most popular permutation-equivariant building blocks for neural network composition are message passing and self-attention.

**Message passing.** In this work we consider message passing on point clouds. Message passing layers act on local neighbourhoods  $\mathcal{N}_p$  around points  $p$  in the input vector space by constructing messages and updating point features  $h^p$

$$m_{p,q} = \text{MLP}(h^p \| h^q \| p \| q), \quad (\text{message from } q \text{ to } p)$$

$$h^p \leftarrow \bigoplus_{q \in \mathcal{N}_p} m_{p,q} \quad (\text{point feature update})$$

where  $\bigoplus$  is an aggregation operator like mean or maximum and  $\|$  denotes concatenation. Permutation equivariance is enabled by the dependence of the local neighbourhoods  $\mathcal{N}_p$  on the input vector space.

**Self-attention.** Given an  $(n \times c_{\text{in}})$ -dimensional tensor  $x$ , multi-head self-attention layers can be defined as follows:

$$a_i = \text{Softmax} \left( \frac{q_i(x) k_i(x)^T}{\sqrt{c_{\text{in}}}} \right) v_i(x)$$

$$A = x + \text{Linear}(\|_i a_i)$$

$$x \leftarrow A + \text{MLP}(A),$$

where Softmax acts entry-wise and we require query, key and value functions  $q_i, k_i, v_i$  as well as Linear and MLP to be row-wise permutation-equivariant. We denote by  $\|_i$  sequential concatenation. Self-attention is a composition of permutation-equivariant operations and computes pair-wise interactions between all  $n$  set elements.

### 2.2.3. E(3)-equivariant neural networks

Apart from being equivariant under permutations of set elements, neural networks can be equivariant under element-wise rotation, translation and reflection in the input vector space  $\mathbb{R}^3$  if they satisfy the condition

$$(f_{\theta} \circ \rho)(\cdot) \equiv (\rho \circ f_{\theta})(\cdot) \quad \text{for all } \rho \in \text{E}(3).$$

In other words, the relation between input and output elements cannot depend on their specific location and orientation in ambient space. All possible rotations, translations and reflections constitute the Euclidean group  $\text{E}(3)$ .

### 2.2.4. DeepONet and MIONet

DeepONet [28] and its extension MIONet [42] are frameworks for learning operators between function spaces. Let  $\gamma$  be a source and  $y$  be a target function. The objective is to learn an operator  $G_{\gamma} : \mathbb{R}^{c_{\text{in}}} \rightarrow \mathbb{R}^{c_{\text{out}}}$  so that  $G_{\gamma}(x) \equiv y(x)$ . MIONet generalises this to multiple source functions which may include boundary conditions  $\xi$ , i.e.,

$$G_{\gamma, \xi}(x) \equiv y_{\xi}(x).$$

Learning this operator is realised by feeding observations of the input functions to “branch” networks  $b_1, b_2$  and coordinates  $x$  to a “trunk” network  $\tau$  and multiplying their outputs entry-wise:

$$G_{\gamma, \xi}(x) = \text{Sum} \left( b_1(\{\gamma^i\}_{i \in [1, m]}) \odot b_2(\{\xi^i\}_{i \in [1, l]}) \odot \tau(x) \right)$$

where Sum denotes summations of all vector entries and  $\odot$  is the Hadamard product. Optionally, a bias term can be added.

### 2.3. Deep vectorised operator

In this work, we propose a model that takes as inputs steady-state hemodynamic fields as well as pulsatile boundary conditions and estimates the pulsatile hemodynamic fields. Let  $\Omega$  be the artery and denote by  $u, p$  velocity and pressure, respectively. Define the steady-state hemodynamic fields as

$$\gamma : \begin{cases} \Omega \rightarrow \mathbb{R}^3 \times \mathbb{R} \\ x \mapsto u \| p \end{cases}$$

and the pulsatile hemodynamic fields as

$$y_{\xi} : \begin{cases} T \times \Omega \rightarrow \mathbb{R}^3 \times \mathbb{R} \\ t, x \mapsto u_{\xi} \| p_{\xi}, \end{cases} \quad \xi \in \Xi$$

where  $\Xi$  is the function space of all pulsatile boundary conditions described by three quantities: velocity, pressure at the artery inlet and pulsatile waveform. We allow  $\xi$  to be dependent on  $t$  which we omit for compactness of notation. Conceptually, our model is in between MIONet and (vectorised) conditional neural fields. Our objective is to learn an operator

$$G_{\gamma, \xi} : T \times \Omega \rightarrow \mathbb{R}^3 \times \mathbb{R} \quad \text{so that} \quad G_{\gamma, \xi}(x) \equiv y_{\xi}(x).$$

We do so via a conditional neural field  $f_{\theta}$  so that point-wise

$$G_{\gamma, \xi}(t, x) = f_{\theta}(\gamma(x) \| t \| r(x) \| \xi)$$

where  $r(x)$  is fixed, relational encoding of coordinates which we describe below. We train  $f_{\theta}$  on tuples of observations

$$\{(\gamma(x^j), t^i, x^j, y_{\xi}(t^i, x^j)) \mid i \in [1, m], j \in [1, n]\}$$

in a vectorised manner. Thus, we can introduce spatial context by allowing  $f_{\theta}$  to be permutation-equivariant. This setup can also be viewed as a time-continuous, space-discrete, conditional neural field.

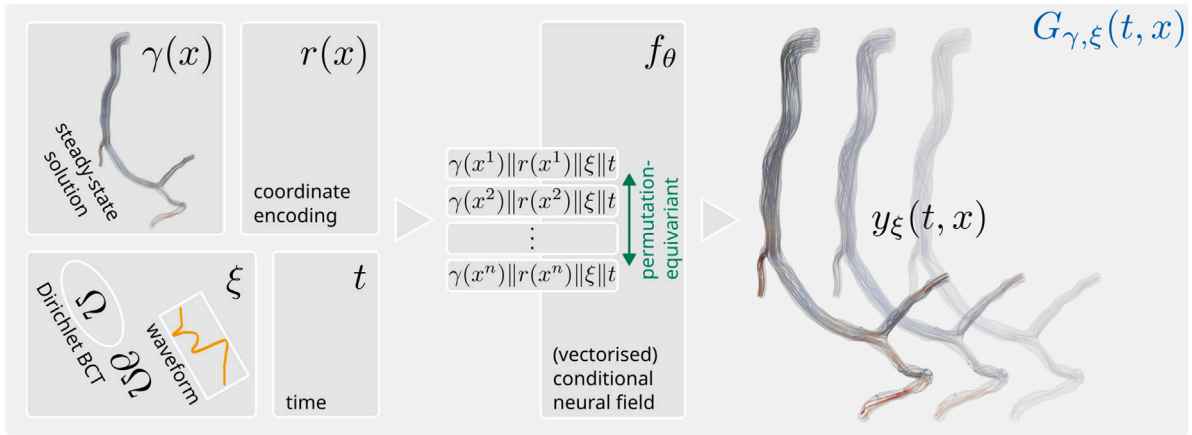
#### 2.3.1. Backend variants

Conditional (vectorised) neural field  $f_{\theta}$  can be embodied by many different neural architectures. When designing neural networks for large-scale hemodynamic problems, it is imperative that the models are scalable in the number of query points w.r.t. computation complexity and device usage. We choose a simple, point-wise MLP, the permutation-equivariant PointNet++ and the permutation- as well as SE(3)-equivariant LaB-GATr [12,43] with memory-efficient attention [44]. We also include LaB-VaTr, which uses the same layout as LaB-GATr but vanilla fully-connected layers instead of geometric algebra.

#### 2.3.2. Coordinate encoding

We equip our model with a powerful coordinate space by lifting  $x^j$  to a higher-dimensional, relational encoding  $r(x^j)$  of spatial location.<sup>2</sup> To this end, we compute – for every point  $x^j$  within the artery – relative position to the inlet, closest outlet and inner lumen wall [13]. We separate these vectors into magnitude and direction, both of which we feed to  $f_{\theta}$ . Furthermore, we employ a geometry-aware location descriptor in the form of *diffusion distance* and direction to inlet and outlets. For this we use the heat method for distance computation [46], originally developed for geodesic distance on surfaces, which readily transfers to volumetric point clouds. The heat method works with a point cloud Laplacian to define an analogue to the heat equation. We can then simulate diffusion of heat sources placed at the inlet and outlets, respectively, and treat the resulting heat map as distance field. Solving the heat equation on point clouds requires solution of a sparse linear system which can be done in near-linear time [46]. To obtain the “direction” of diffusion, we simply take the gradient of said field via the discretisation scheme described in [47].

<sup>2</sup> Despite its similar name, the purpose of this coordinate encoding is to construct canonical coordinates across arteries, different from positional encoding for combatting spectral bias [45].



**Fig. 2.** Deep vectorised operator. Steady-state hemodynamic fields  $\gamma(x)$  and relational point encoding  $r(x)$  are fed to a permutation-equivariant model  $f_\theta$  together with conditions  $\xi$  and query time  $t$ . Functional quantities  $\gamma(x)$  and  $r(x)$  can be vectorised, i.e., supplied all at once, to enable cross-conditioning by parametrising  $f_\theta$  with message passing or self-attention layers. In that case, global quantities  $\xi$  and  $t$  are broadcast over the batch dimension. Neural field  $f_\theta$  then predicts pulsatile hemodynamics  $y_\xi(t, x)$  at (each of) the query point(s)  $(t, x)$  in spacetime. This defines the nonlinear operator  $G_{\gamma, \xi}(t, x)$ .

### 2.3.3. Boundary condition encoding

As conditions  $\xi$ , we supply to our model at each point in time  $t$  the average velocity vector over the inlet, decomposed into magnitude and direction, along with the average pressure over the inlet. We assume access to both these quantities because they can readily be measured in hospitals. Additionally, we assume access to a patient-specific blood flow waveform of which we query the mean, standard deviation, minimum and maximum value over the cardiac cycle, which we feed to our model along with the subject’s heart rate. We do so by simple concatenation to the point-wise observations of the functional quantities  $\gamma(x)$  and  $r(x)$  (compare Fig. 2).

### 2.3.4. Pressure drop estimation

Instead of estimating point-wise pressure  $p(t, x)$  directly, we found it beneficial to predict the *pressure drop* relative to the pressure at the inlet.

### 2.4. Evaluation metrics

To evaluate our method, we must compare vector fields, i.e., point-wise model predictions  $f_\theta(\dots) =: f_\theta^p$  and ground truth  $y_\xi(\dots) =: y_\xi^p$ . We do so via the following metrics. We define approximation disparity

$$\text{Approx. disp.} : \sqrt{\sum_p \|y_\xi^p - f_\theta^p\|_2^2 / \sum_p \|y_\xi^p\|_2^2}$$

which measures the similarity between the vector fields. Furthermore, we use mean cosine similarity between vectors

$$\text{Cos. similarity} : \text{mean} \cos \angle(y_\xi^p, f_\theta^p)$$

which ranges between  $-1$  (opposite) and  $1$  (proportional) and measures directional agreement independent of magnitude.

## 3. Experiments and results

### 3.1. Implementation

We implemented all neural networks in Python using PyTorch [48] and PyTorch Geometric [49]. For PointNet++, we opted for an aggressive sampling strategy with cumulative factors of 0.0033, 0.3344, 0.6656 and 0.9967 to account for the large-scale data. Likewise, we chose a compression ratio of 0.0033 for LaB-GATr and LaB-VaTr. We let LaB-GATr use cross-attention for the compression module and LaB-VaTr message passing. We chose these settings since they resulted in favourable runtime and accuracy. We chose hidden channels size so

all networks had approximately 0.65 million trainable parameters. We used PyTorch just-in-time compilation to speed up training for MLP while the other models did not benefit from it due to unsupported software.

### 3.2. Training details

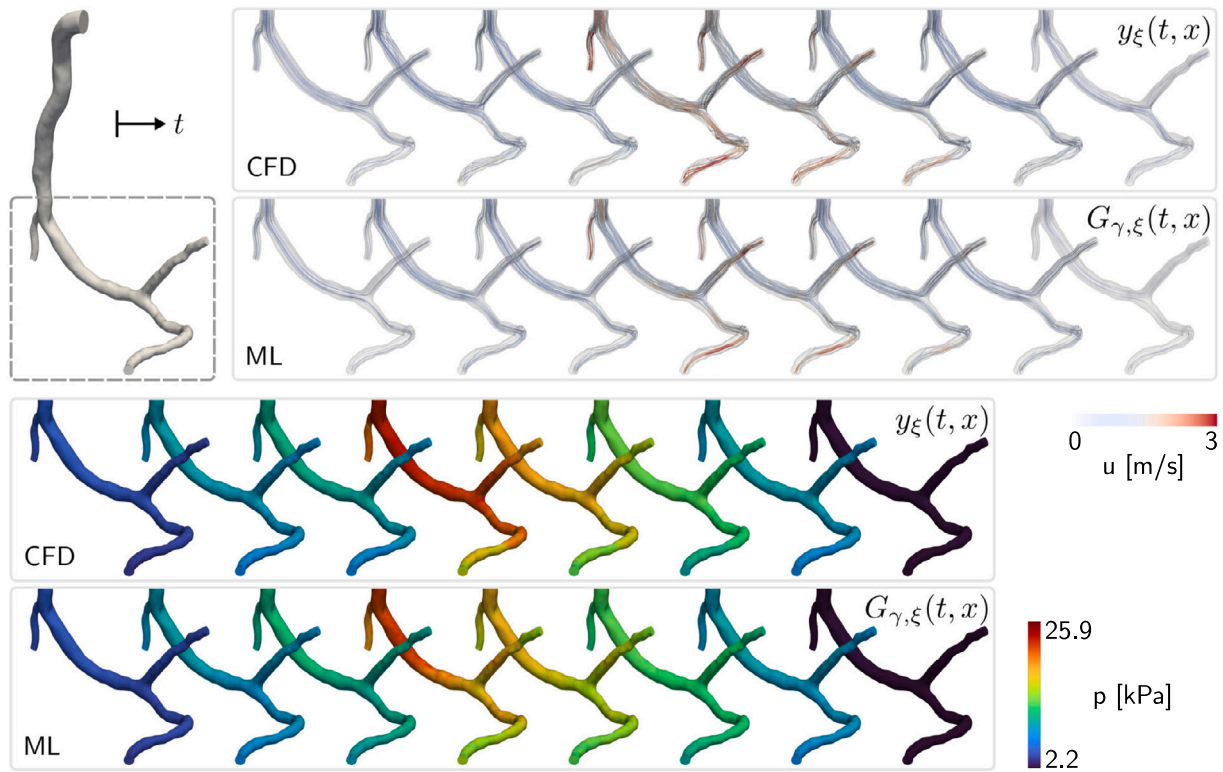
We split the 74 patient-specific CFD simulations into 55 training, 4 validation and 15 test cases. We ensured that training and test split had the same ratio between left and right coronary arteries. Left and right coronary artery of the same subject were considered separate cases. Where the dataset contained geometries of the left as well as the right coronary artery of the same subject, both where included in the same dataset split. All models were trained with batch size 12 (enabled by gradient accumulation) on NVIDIA L40 (48 GB) GPUs using Adam optimiser (learning rate  $3 \cdot 10^{-4}$ ). We trained MLP, PointNet++ and LaB-VaTr each for 1000 epochs on a single GPU which took 56:28 h on average. We trained LaB-GATr for 200 epochs (47:00 h) on four parallel GPUs to ensure fair comparison. For each neural network, we performed two training runs, with and without (both) exponential learning rate decay and gradient clipping. LAB-GATr showed increased accuracy while the other models did not benefit from it. Our implementation is publicly available.<sup>3</sup>

### 3.3. Hemodynamics estimation

Fig. 3 compares the estimated pulsatile hemodynamics by our deep vectorised operator  $G_{\gamma, \xi}$  in a subject from the test split to CFD. Here,  $G_{\gamma, \xi}$  uses LaB-GATr as backend. From the streamlines, we see good overall agreement between ground truth and prediction. We observe the highest disparity in regions with high-frequency flow components,<sup>4</sup> e.g., around the bifurcation and along the tortuous downstream vasculature during peak systole (time steps four and five). Looking at the pressure maps, we observe excellent agreement between  $G_{\gamma, \xi}$  and CFD in upstream vasculature which slightly and gradually decreases moving downstream (compare again peak systole). This might be because we prescribe boundary conditions  $\xi$  exclusively at the artery inlet in  $G_{\gamma, \xi}$ , since it is easier to measure in a clinical scenario. We put the qualitative evaluation into perspective by reporting mean absolute error (MAE) between estimation and ground truth across the test split in Table 2.

<sup>3</sup> [github.com/sukjulian/deep-vectorised-operators](https://github.com/sukjulian/deep-vectorised-operators).

<sup>4</sup> By high-frequency, we mean high variation in direction and magnitude on a relatively small spatial and temporal scale.



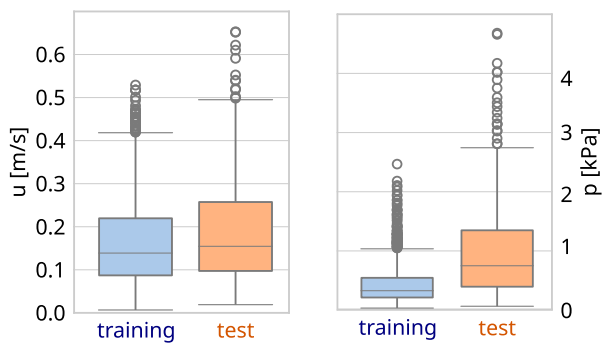
**Fig. 3.** Qualitative evaluation of machine learning (ML) model  $G_{\gamma,\xi}(t,x)$  in comparison to CFD  $y_{\xi}(t,x)$  in a test subject (generalisation). Approx. disp. associated with  $u||p$  estimation in this subject was 0.325. We visualise streamlines and pressure maps for eight time points in the cardiac cycle with spacing  $dt = 0.125$  s. The backend model was LaB-GATr.

**Table 1**

Quantitative evaluation of backends. We report mean  $\pm$  standard deviation (across time and subjects) of Approx. disp. and Cos. similarity w.r.t. hemodynamics  $u||p$  jointly, as well as velocity  $u$  and pressure  $p$  separately. We denote in **bold** best evaluation metrics where they significantly ( $p < 0.05$ ) differ from the next best (one-way ANOVA test). Additionally, we state each model's training complexity.

Backend	Train. complex.	$u  p$		$p$	
		Approx. disp. $\downarrow$	Approx. disp. $\downarrow$	Cos. similarity $\uparrow$	Approx. disp. $\downarrow$
MLP	254 s/epoch	0.432 $\pm$ 0.065	0.439 $\pm$ 0.066	0.72 $\pm$ 0.05	0.121 $\pm$ 0.068
PointNet++	164 s/epoch	0.382 $\pm$ 0.073	0.388 $\pm$ 0.073	<b>0.76</b> $\pm$ 0.04	0.115 $\pm$ 0.075
LaB-GATr	857 s/epoch <sup>a</sup>	<b>0.368</b> $\pm$ 0.079	<b>0.374</b> $\pm$ 0.080	<b>0.76</b> $\pm$ 0.04	<b>0.105</b> $\pm$ 0.049
LaB-VaTr	191 s/epoch	0.402 $\pm$ 0.076	0.409 $\pm$ 0.076	0.74 $\pm$ 0.04	<b>0.098</b> $\pm$ 0.072

<sup>a</sup> On four devices in parallel.



**Fig. 4.** Distribution of MAE  $\downarrow$  (over time and subjects) between the machine learning model compared to CFD as boxplots for both training and test split for estimation of velocity  $u$  and pressure  $p$ , respectively. The backend model was LaB-GATr.

We find that MAE is an order of magnitude lower than the values observed in Fig. 3. Furthermore, in Fig. 4 we provide comparison of the distribution of MAE across both training and test split. Fig. 5 shows three levels of accuracy – worst, average and best – among time steps

**Table 2**

Mean absolute error (MAE)  $\downarrow$  of the machine learning model compared to CFD. We associate each time step in each test subject with a scalar MAE value and report the average and maximum among all of these. The backend model was LaB-GATr.

$u$ [m/s]		$p$ [kPa]		vFFR	
Average	Max.	Average	Max.	Average	Max.
0.189	0.653	1.0	4.7	0.07	0.25

in all subjects. We chose velocity  $u$  for this comparison. In the average case, we find good agreement, even for difficult regions around the bifurcations. In the worst case, which is associated with peak systole, we observe difficulties capturing the high-frequency flow components which supports the observations above. In Fig. 6 we present the spatial distribution of MAE at each point in the cardiac cycle for three subjects in the test set, corresponding to three levels of accuracy w.r.t. estimation of velocity  $u$ . We observe that the highest errors, both for velocity and pressure estimation, occur during peak systole in all three cases. The amplitude of the boxplots can be interpreted as spatial deviation of MAE. For example, the error is higher in arterial bifurcations (compare Fig. 5). We observe that during peak systole, spatial deviation of MAE increases, which can be explained by formation of high-frequency flow components.

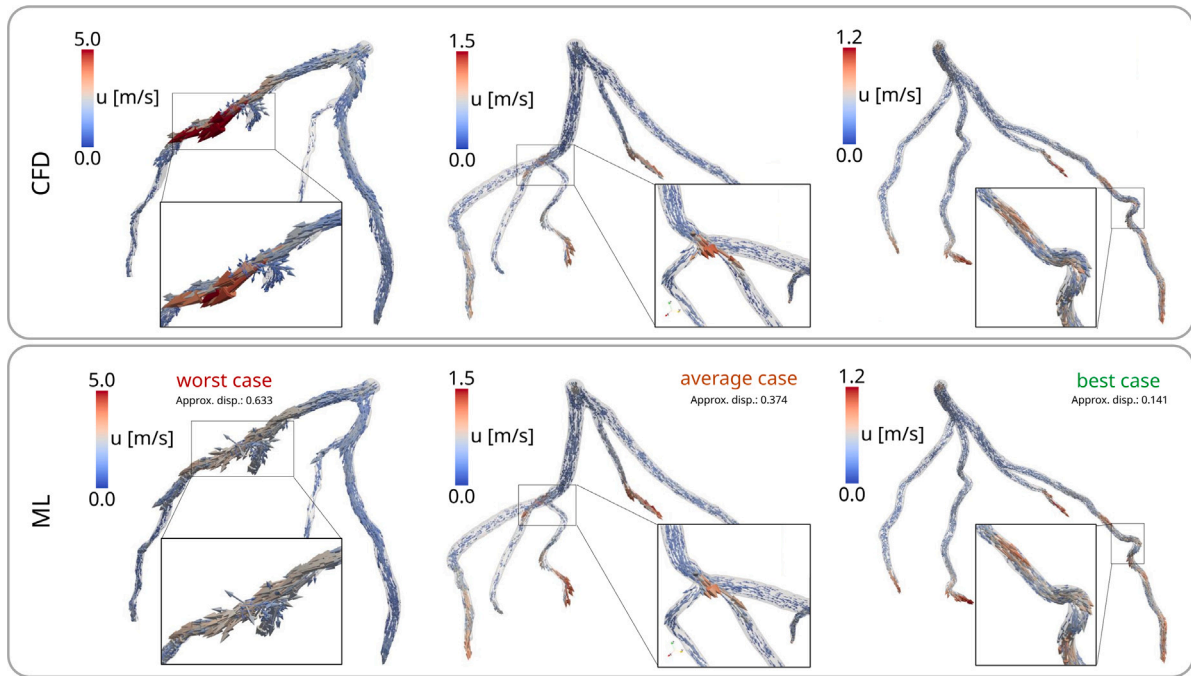


Fig. 5. Comparative study of different levels of accuracy (w.r.t. Approx. disp. associated with  $u$ ) of the machine learning (ML) model in comparison to CFD in different test subjects (generalisation). We visualise velocity field vectors corresponding to isolated points in time. The backend model was LaB-GATr.

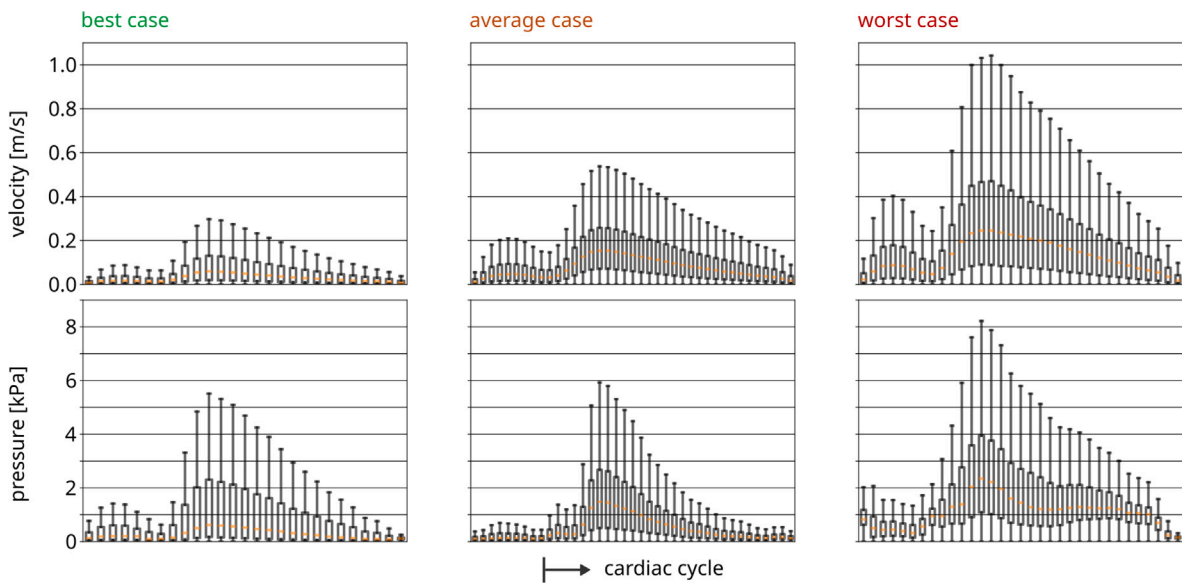


Fig. 6. MAE ↓ over cardiac cycle for test subjects (generalisation) corresponding to different levels of accuracy (w.r.t. time-averaged MAE associated with  $u$ ). Distribution of MAE over spatial points in the artery is shown as boxplot for each temporal point in the cardiac cycle. We exclude outliers for visual clarity (due to the very large number of spatial points). The backend model was LaB-GATr.

### 3.4. Estimated fractional flow reserve

We evaluate the capability of our machine learning model to estimate “virtual” fractional flow reserve (vFFR), which we define as pressure field  $p$ , divided point-wise by the average pressure over the artery inlet. In Table 2 we compare estimated vFFR between our deep vectorised operator and CFD on the basis of MAE across the test split. We find that average MAE is as low as 0.07 while maximum MAE among all time steps in all subjects breaches the clinical decision threshold (0.75–0.8) at 0.25. Note that this evaluation is based on the volumetric pressure field rather than a surface map (the latter of which is sufficient for vFFR) and no task-specific fine-tuning was performed.

### 3.5. Backend study

In Table 1 we compare model predictions on the held-out test split to the corresponding ground truth values via Approx. disp. and Cos. similarity. LaB-GATr achieves the highest accuracy on the estimation of  $u||p$ , followed by PointNet++, LaB-VaTr and MLP with statistically significant ( $p < 0.05$  in a one-way ANOVA test) difference. When isolating the target fields, PointNet++ and LaB-GATr are tied for most accurate (directional) velocity prediction in terms of Cos. similarity while LaB-GATr achieves lower Approx. disp.. LaB-VaTr achieves the lowest mean Approx. disp. in pressure field estimation, however, there is no statistically significant difference to LaB-GATr. In terms of training

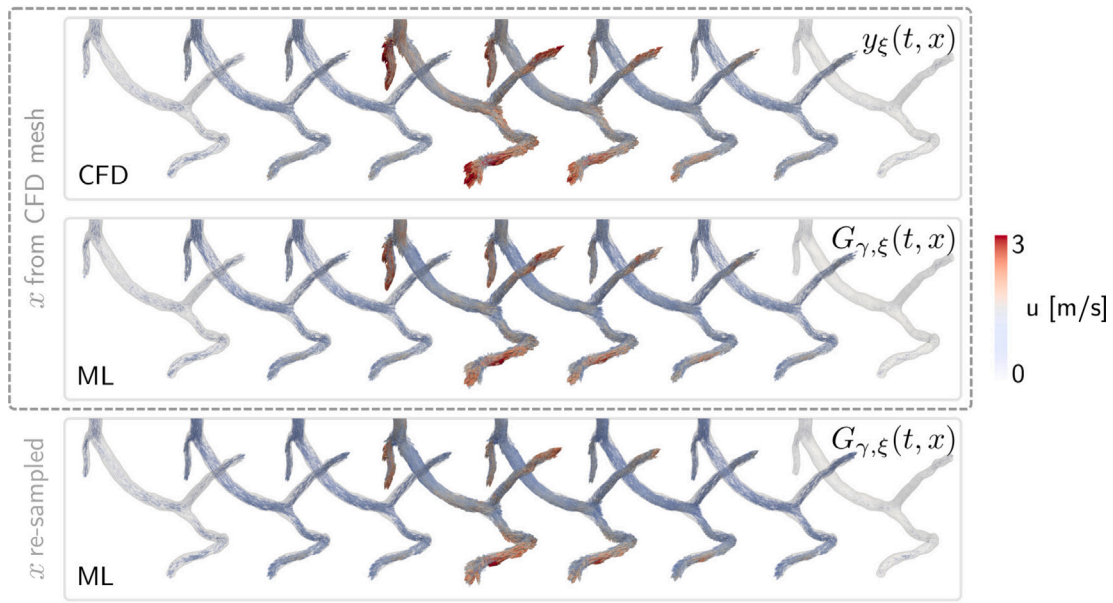


Fig. 7. Discretisation sensitivity analysis of machine learning (ML) model in a test subject (generalisation). We evaluate  $G_{\gamma, \xi}(t, x)$  on positions  $x$  induced by the CFD mesh as well randomly sampled positions from the domain and visualise the resulting velocity fields. The backend model was LaB-GATr.

Table 3

Multitask versus isolated training target. We re-trained PointNet++ three times to predict only velocity  $u$ , only pressure  $p$  and both  $u||p$  jointly and report the resulting metrics.

Training target	$u$		$p$
	Approx. disp. ↓	Cos. similarity ↑	Approx. disp. ↓
$u  p$	<b>0.386</b> ± 0.072	<b>0.76</b> ± 0.04	<b>0.114</b> ± 0.073
$u$	<b>0.388</b> ± 0.074	<b>0.76</b> ± 0.04	0.345 ± 0.148
$p$	1.006 ± 0.018	-0.16 ± 0.12	<b>0.113</b> ± 0.068

complexity, PointNet++ is the fastest at 164 s per epoch, outpacing MLP despite just-in-time compilation. LaB-GATr is computationally heavy to train, requiring 857 s per epoch even when parallelised across four GPUs.

### 3.6. Multitask versus isolated target learning

We investigate the possible advantage of isolating the estimation of velocity and pressure into separately trained neural networks. To this end, we re-trained a model with the PointNet++ backend for 400 epochs three times, with prediction target  $u||p$  (jointly), as well as  $u$  and  $p$  (isolated). In Table 3 we present the results. We did not find statistically significant ( $p < 0.05$  in a one-way ANOVA test) difference between multitask and isolated target learning, indicating that multitask learning neither benefits nor harms prediction of velocity or pressure fields.

### 3.7. Sensitivity to discretisation

We investigate our models' capability for zero-shot generalisation to re-discretisation of the domain. To this end, we discard the mesh vertices and sample new positions (freely in space), approximately evenly from the surface of the artery and from within the artery lumen. We make sure the number of sampled surface and volume points are approximately equal to the CFD meshes used during training. We then evaluate the pre-trained (vectorised) conditional neural field  $f_{\theta}$  with MLP, PointNet++, LaB-GATr and LaB-VaTr as backends on these newly sampled points. To enable comparison to the ground truth, we perform proportional interpolation of steady state and pulsatile hemodynamic fields  $\gamma, \gamma_{\xi}$  based on distance to four closest points in

Table 4

Discretisation sensitivity. We evaluate the pre-trained models on new point coordinates that are randomly sampled from the domain. We report the difference  $\Delta$  in Approx. disp. for joint prediction of  $u||p$  and check whether the difference is statistically significant (one-way ANOVA test).

Backend	$u  p$ $\Delta$ Approx. disp. ↓
MLP	-0.004
PointNet++	+0.021*
LaB-GATr	-0.000
LaB-VaTr	-0.003

\* Significant ( $p < 0.05$ ).

the source mesh. In Table 4 we report  $\Delta$  Approx. disp. (the difference in Approx. disp.) between training resolution and randomly sampled points in subjects from the test split. Positive values mean that Approx. disp. increased, i.e., accuracy decreased. We find that for all four backends  $\Delta$  Approx. disp. is marginal. Furthermore, we find that for MLP, LaB-GATr and LaB-VaTr, the difference is not statistically significant ( $p > 0.05$  in a one-way ANOVA test), i.e., re-discretisation has not changed their accuracy. The PointNet++ backend is the only one resulting in statistically significant  $\Delta$  Approx. disp.. Fig. 7 showcases the discretisation-robustness of (vectorised) conditional neural field  $f_{\theta} = G_{\gamma, \xi}$  for the example of velocity field estimation. Here,  $f_{\theta}$  uses LaB-GATr as backend. We observe that the vector fields resulting from evaluation of  $G_{\gamma, \xi}(t, x)$  on positions  $x$  from the CFD mesh (training discretisation) and randomly sampled positions visually coincide. The amount of outliers is negligible.

## 4. Discussion and conclusion

In this work, we presented a novel deep learning approach that takes the end-diastolic steady-state hemodynamics in the coronary arteries and estimates the corresponding pulsatile hemodynamics, over the full cardiac cycle. This idea is based on the observation that steady-state simulations are orders of magnitude faster than pulsatile simulations while medically relevant biomarkers may be related to pulsatile flow, e.g., multi-directional wall shear stress [50]. Our model is conditioned on the patient-specific pulsatile waveform, as well as the flow and pressure over the artery inlet. All of these quantities

are easily accessible from direct or indirect clinical measurements. We evaluated our method on a dataset of CFD-based hemodynamic simulations in stenotic left and right coronary arteries encompassing 74 cases across a cohort of real-life patients. By casting the learning objective as operator learning from steady-state to pulsatile flow, our method is able to accurately capture the 4D nature of blood flow based on limited amounts of (fairly diverse) data. Steady-state simulations were 24 to 288 times faster than pulsatile simulations in these subjects. Consequently, our approach could potentially be useful in the clinic to make pulsatile hemodynamic fields available in settings where time is critical or computational resources are limited. Medical decision markers like wall shear stress, oscillatory shear index and fractional flow reserve can then be computed from pressure and velocity fields. Importantly, our method is robust w.r.t. discretisation of the flow domain which could be beneficial for inter-operator variability.

We described this approach in a unified framework which we called deep vectorised operator, since our model maps between infinite-dimensional function spaces. Deep vectorised operators are parametrised by (vectorised) conditional neural fields [41] that can be embodied by several backends, as long as they are point-wise permutation-equivariant like message passing or self-attention models. Importantly, *vectorised* neural fields differ from neural fields in the sense that all query points are fed to the model at once, in a vectorised way. Our empirical evidence justifies the nomenclature “neural field” by suggesting that deep vectorised operators are discretisation-independent, i.e., enable zero-shot generalisation to re-sampling of the domain. This is remarkable, because the neural architectures we tested, in particular PointNet++ [9], are based on cross-conditioning between the query points. Our evidence suggests that they nevertheless function as a mapping between domain and co-domain, just like traditional neural fields [37] parametrised by point-wise neural architectures like MLP. As expected, the deep vectorised operator with MLP as backend was agnostic to re-sampling of the domain while PointNet++ exhibited small (but statistically significant) sensitivity, hinting at pronounced cross-conditioning.

As data-driven approach, the accuracy of the presented method is proportional to the amount of available data. In our evaluations, we observed loss of accuracy for high-frequency flow components. This indicates that the model is not able to reliably overcome spectral bias [45] based on the available amount of data. However, we expect improvements when more training data is available. While the presented method maps between function spaces and is thus agnostic to the topology (e.g. number of bifurcations) and discretisation (e.g. number of mesh vertices) of the domain, its estimations will only be accurate insofar the training data sufficiently resembles the data seen at inference time.<sup>5</sup> This argument extends to the encoding of Dirichlet boundary conditions and waveform. The operator learning setup affords representational freedom, e.g., discretisation of any (including real-world) waveform at a fixed number of points in time. Since our training data was based on a scaled template waveform from the literature, we opted for simple statistical measures, though. Ultimately, more sophisticated waveforms would require a larger amount of representative training data, of leveraging which – we argue – our model is a-priori capable. Investigating applicability and adaptability to realistic, real-world waveforms is an important avenue for future work.

Our approach is similar in scope to DeepONet and MIONet as we are learning a deep operator between function spaces. In comparison to MIONet, we sacrificed the entry-wise Hadamard-product structure for scalability and ease of implementation. This affects the learned operator and in turn the available mathematical approximation theory, investigating which is an interesting avenue for future work. In comparison

<sup>5</sup> In particular, this means that for clinical adoption, each hospital would have to train its own model on proprietary data or a large representative dataset would have to be collected to train a universal foundation model.

to Maul et al. [24], our work explores the incorporation of steady-state hemodynamics as powerful prior for pulsatile hemodynamics, our model returns velocity *and* pressure and our dataset is composed of real-life patients. For future work, we plan to incorporate a balance term for inlet and outlet flux in the training loss, as well as continuity and momentum terms, to improve consistency with the flow physics and pressure estimation at the artery outlet. A key limitation in the present study is that, while medical decision markers like wall shear stress, oscillatory shear index and (virtual) fractional flow reserve can indeed be computed from deep vectorised operators’ estimated velocity and pressure fields, we only showcase this for vFFR. Since wall shear stress (and consequently oscillatory shear index) is a first-order differential quantity which depends on the Jacobian of the velocity field, there is no naive way of computing it and we require interpolation, finite differences or finite elements. In order for deep vectorised operators to output sufficiently smooth velocity fields, we conjecture that at least first-order regularisation is required during training. In detail, we aim to include a Neumann boundary condition relating the near-wall velocity field with the (finite-elements-based) wall shear stress from CFD. As this approach requires differentiation, it blends in with the aforementioned continuity and momentum regularisation. We plan on investigating these jointly.

In conclusion, deep vectorised operators can be used to estimate pulsatile from steady-state hemodynamics in coronary arteries while being independent of the discretisation of the domain. Furthermore, deep vectorised operators are a useful modelling tool with many expected applications in cardiovascular medicine and biomedical engineering.

#### CRediT authorship contribution statement

**Julian Suk:** Validation, Conceptualization, Visualization, Methodology, Formal analysis, Writing – original draft, Software, Investigation. **Guido Nannini:** Validation, Writing – review & editing, Methodology, Conceptualization, Data curation. **Patryk Rygiel:** Validation, Formal analysis, Writing – review & editing, Conceptualization, Methodology. **Christoph Brune:** Methodology, Supervision, Formal analysis, Writing – review & editing, Project administration. **Gianluca Pontone:** Data curation, Writing – review & editing, Resources, Funding acquisition, Validation, Project administration. **Alberto Redaelli:** Writing – review & editing, Resources, Methodology, Validation, Data curation. **Jelmer M. Wolterink:** Validation, Resources, Investigation, Formal analysis, Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

#### Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Julian Suk reports financial support was provided by 4TU Precision Medicine. Jelmer M. Wolterink reports financial support was provided by NWO domain Applied and Engineering Sciences. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

This work is funded in part by the 4TU Precision Medicine programme supported by High Tech for a Sustainable Future, a framework commissioned by the four Universities of Technology of the Netherlands. Jelmer M. Wolterink was supported by the NWO domain Applied and Engineering Sciences VENI grant (18192).

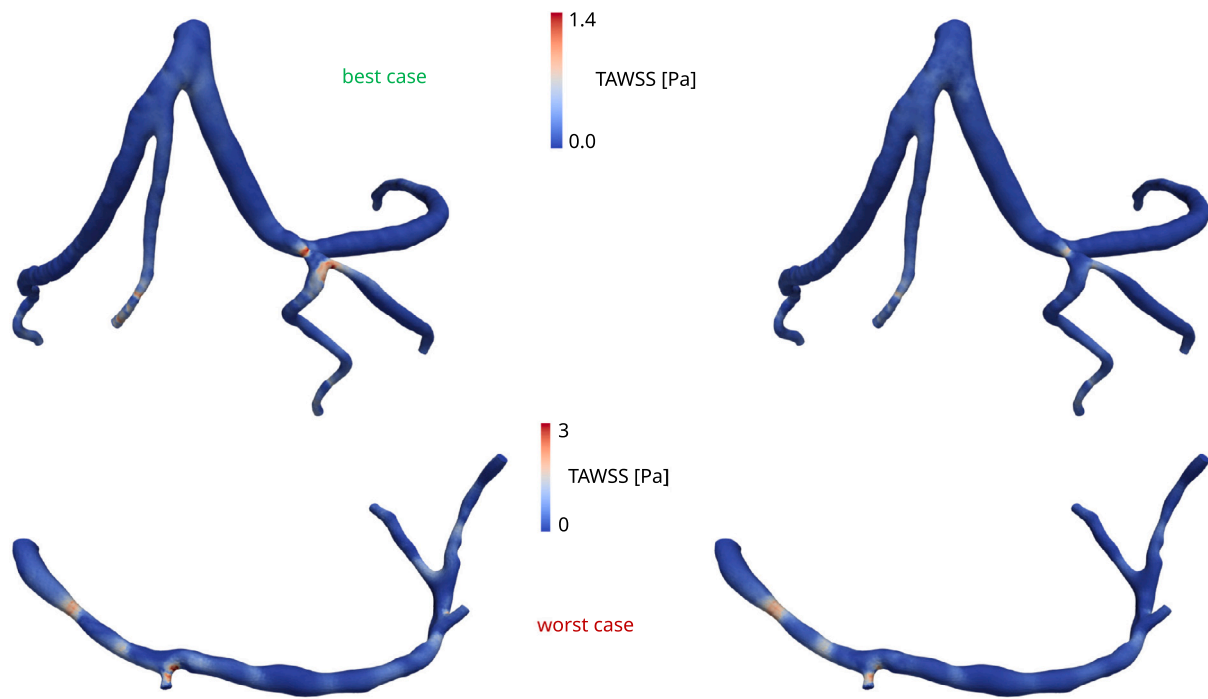


Fig. 8. Preliminary results on time-averaged wall shear stress (TAWSS) extraction from the estimated velocity field at different levels of accuracy (w.r.t. MAE associated with TAWSS) in test subjects. The backend model was LaB-GATr.

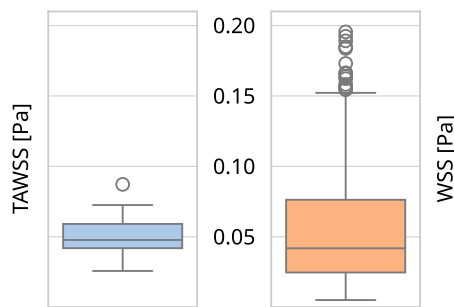


Fig. 9. Distribution of MAE ↓ of time-averaged wall shear stress (TAWSS) (over subjects) and wall shear stress (WSS) (over time and subjects) between the machine learning model compared to CFD as boxplots for the test split. The backend model was LaB-GATr.

Appendix A. Preliminary results on wall shear stress

In Fig. 8 we show preliminary results on time-averaged wall shear stress (TAWSS) extraction from the estimated velocity fields for two levels of accuracy. We compute wall shear stress (WSS) via  $L^2$  projection based on the tetrahedral mesh from CFD simulation. We observe that high TAWSS is captured in terms of artery segment but not magnitude. We attribute this to the fact that wall shear stress is a first-order differential quantity for which we do not yet account for via velocity field regularisation during training. In Fig. 9 we provide comparison of the distribution of MAE for TAWSS and WSS across the test split.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.cmpb.2025.108958>.

References

[1] A. Zingaro, C. Vergara, L. Dede, F. Regazzoni, A. Quarteroni, A comprehensive mathematical model for cardiac perfusion, *Sci. Rep.* 13 (2023).

[2] R.S. Driessen, I. Danad, W.J. Stuijzand, P.G. Raijmakers, S.P. Schumacher, P.A. van Diemen, J.A. Leipsic, J. Knuuti, S.R. Underwood, P.M. van de Ven, A.C. van Rossum, C.A. Taylor, P. Knaapen, Comparison of coronary computed tomography angiography, fractional flow reserve, and perfusion imaging for ischemia diagnosis, *J. Am. Coll. Cardiol.* 73 (2) (2019) 161–173.

[3] A. Candrea, M. Pagnoni, M.L. Rizzini, T. Mizukami, E. Gallinoro, V. Mazzi, D. Gallo, D. Meier, T. Shinke, J.-P. Aben, S. Nagumo, J. Sonck, D. Munhoz, S. Fournier, E. Barbato, W. Heggermont, S. Cook, C. Chiastra, U. Morbiducci, B. De Bruyne, O. Muller, C. Collet, Risk of myocardial infarction based on endothelial shear stress analysis using coronary angiography, *Atheroscler.* 342 (2022) 28–35.

[4] B. Chung, J.R. Cebal, CFD for evaluation and treatment planning of aneurysms: review of proposed clinical uses and their challenges, *Ann. Biomed. Eng.* 43 (1) (2015) 122–138.

[5] J.M. Lee, G. Choi, B.-K. Koo, D. Hwang, J. Park, J. Zhang, K.-J. Kim, Y. Tong, H.J. Kim, L. Grady, J.-H. Doh, C.-W. Nam, E.-S. Shin, Y.-S. Cho, S.-Y. Choi, E.J. Chun, J.-H. Choi, B.L. Nørgaard, E.H. Christiansen, K. Niemen, H. Otake, M. Penicka, B. de Bruyne, T. Kubo, T. Akasaka, J. Narula, P.S. Douglas, C.A. Taylor, H.-S. Kim, Identification of high-risk plaques destined to cause acute coronary syndrome using coronary computed tomographic angiography and computational fluid dynamics, *JACC: Cardiovasc. Imaging* 12 (6) (2019) 1032–1043.

[6] G. Nannini, S. Saitta, L. Mariani, R. Maragna, A. Baggiano, S. Mushtaq, G. Pontone, A. Redaelli, An automated and time-efficient framework for simulation of coronary blood flow under steady and pulsatile conditions, *Comput. Methods Programs Biomed.* 257 (2024) 108415.

[7] K. Valen-Sendstad, A. Bergersen, Y. Shimogonya, L. Goubergrits, J. Bruening, J. Pallares, S. Cito, S. Piskin, K. Pekkan, A. Geers, I. Larrabide, S. Rapaka, V. Mihalef, W. Fu, A. Qiao, K. Jain, S. Roller, K.-A. Mardal, R. Kamakoti, T. Spirka, N. Ashton, A. Revell, N. Aristokleous, J. Houston, M. Tsuji, F. Ishida, P. Menon, L. Browne, S. Broderick, M. Shojima, S. Koizumi, M. Barbour, A. Aliseda, H. Morales, T. Lefevre, S. Hodis, Y. Al-Smadi, J. Tran, A. Marsden, S. Vaippummadhom, G. Einstein, A. Brown, K. Debus, K. Niizuma, S. Rashad, S. ichiro Sugiyama, M. Khan, A. Updegrove, S. Shadden, B. Cornelissen, C. Majoie, P. Berg, S. Saalfeld, K. Kono, D. Steinman, Real-world variability in the prediction of intracranial aneurysm wall shear stress: The 2015 international aneurysm CFD challenge, *Cardiovasc. Eng. Technol.* 9 (4) (2018) 544–564.

[8] K. Wu, E. Wu, B. Theodorou, W. Liang, C. Mack, L. Glass, J. Sun, J. Zou, Characterizing the clinical adoption of medical AI devices through U.S. insurance claims, *NEJM AI* 1 (1) (2024) A1oa2300030.

[9] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep hierarchical feature learning on point sets in a metric space, in: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4–9, 2017, Long Beach, CA, USA, 2017, pp. 5099–5108.

- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, December 4-9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.
- [11] X. Morales Ferez, J. Mill, K.A. Juhl, C. Acebes, X. Iriart, B. Legghe, H. Cochet, O. De Backer, R.R. Paulsen, O. Camara, Deep learning framework for real-time estimation of in-silico thrombotic risk indices in the left atrial appendage, *Front. Physiol.* 12 (2021).
- [12] J. Suk, P.D. Haan, B. Imre, J.M. Wolterink, Geometric algebra transformers for large 3D meshes via cross-attention, in: *ICML 2024 Workshop on Geometry-Grounded Representation Learning and Generative Modeling*, 2024, pp. 0–6.
- [13] J. Suk, C. Brune, J.M. Wolterink, SE(3) symmetry lets graph neural networks learn arterial velocity estimation from small datasets, in: O. Bernard, P. Clarysse, N. Duchateau, J. Ohayon, M. Viallon (Eds.), *Functional Imaging and Modeling of the Heart*, Springer Nature Switzerland, Cham, 2023, pp. 445–454.
- [14] D. Lopes, H. Puga, J. Teixeira, R. Lima, Blood flow simulations in patient-specific geometries of the carotid artery: A systematic review, *J. Biomech.* 111 (2020) 110019.
- [15] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, *Sci.* 367 (6481) (2020) 1026–1030.
- [16] M.F. Fathi, I. Perez-Raya, A. Baghaie, P. Berg, G. Janiga, A. Arzani, R.M. D'Souza, Super-resolution and denoising of 4D-Flow MRI using physics-informed deep neural nets, *Comput. Methods Programs Biomed.* 197 (2020) 105729.
- [17] A. Kontogiannis, M. Juniper, Physics-informed compressed sensing: reconstruction of magnetic resonance velocimetry signals as an inverse Navier-Stokes problem, in: *APS Division of Fluid Dynamics Meeting Abstracts*, in: *APS Meeting Abstracts*, 2021, p. H20.001.
- [18] L. Pegolotti, M.R. Pfaller, N.L. Rubio, K. Ding, R. Brugarolas Brufau, E. Darve, A.L. Marsden, Learning reduced-order models for cardiovascular simulations with graph neural networks, *Comput. Biol. Med.* 168 (2024) 107676.
- [19] J. Suk, P. de Haan, P. Lippe, C. Brune, J.M. Wolterink, Mesh neural networks for SE(3)-equivariant hemodynamics estimation on the artery wall, *Comput. Biol. Med.* 173 (2024) 108328.
- [20] R. Gharleghi, A. Sowmya, S. Beier, Transient wall shear stress estimation in coronary bifurcations using convolutional neural networks, *Comput. Methods Programs Biomed.* 225 (2022) 107013.
- [21] P. Rygiel, P. Płuszka, M. Zięba, T. Konopczyński, CenterlinePointNet++: A new point cloud based architecture for coronary artery pressure drop and vFFR estimation, in: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023: 26th International Conference*, Vancouver, BC, Canada, October 8–12, 2023, Proceedings, Part VII, Springer-Verlag, Berlin, Heidelberg, 2023, pp. 781–790.
- [22] L. Liang, W. Mao, W. Sun, A feasibility study of deep learning for predicting hemodynamics of human thoracic aorta, *J. Biomech.* 99 (2020) 109544.
- [23] S. Wang, D. Wu, G. Li, Z. Zhang, W. Xiao, R. Li, A. Qiao, L. Jin, H. Liu, Deep learning-based hemodynamic prediction of carotid artery stenosis before and after surgical treatments, *Front. Physiol.* 13 (2023) 1094743.
- [24] N. Maul, K. Zinn, F. Wagner, M. Thies, M. Rohleder, L. Pfaff, M. Kowarschik, A. Birkhold, A. Maier, Transient hemodynamics prediction using an efficient otree-based deep learning model, in: A. Frangi, M. de Bruijne, D. Wassermann, N. Navab (Eds.), *Information Processing in Medical Imaging*, Springer Nature Switzerland, Cham, 2023, pp. 183–194.
- [25] G. Li, H. Wang, M. Zhang, S. Tupin, A. Qiao, Y. Liu, M. Ohta, H. Anzai, Prediction of 3D cardiovascular hemodynamics before and after coronary artery bypass surgery via deep learning, *Commun. Biol.* 4 (2021).
- [26] X. Zhang, B. Mao, Y. Che, J. Kang, M. Luo, A. Qiao, Y. Liu, H. Anzai, M. Ohta, Y. Guo, G. Li, Physics-informed neural networks (PINNs) for 4D hemodynamics prediction: An investigation of optimal framework based on vascular morphology, *Comput. Biol. Med.* 164 (2023) 107287.
- [27] D. Wessels, D.M. Knigge, R. Valperga, S. Papa, S. Vadgama, E. Gavves, E.J. Bekkers, Grounding continuous representations in geometry: Equivariant neural fields, in: *The Thirteenth International Conference on Learning Representations*, 2025, pp. 0–21.
- [28] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229.
- [29] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: learning maps between function spaces with applications to PDEs, *J. Mach. Learn. Res.* 24 (1) (2024).
- [30] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A.M. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021, pp. 0–16.
- [31] B. Alkin, A. Fürst, S. Schmid, L. Gruber, M. Holzleitner, J. Brandstetter, Universal physics transformers: A framework for efficiently scaling neural operators, in: A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J.M. Tomczak, C. Zhang (Eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024*, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024, pp. 0–16.
- [32] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J.V. Miller, S. Pieper, R. Kikinis, 3D Slicer as an image computing platform for the Quantitative Imaging Network, *Magn. Reson. Imaging* 30 (9) (2012) 1323–1341.
- [33] G. Pontone, S. Moharem-Elgamal, P. Maurovich-Horvat, O. Gaemperli, F. Pugliese, M. Westwood, A. Stefanidis, K.F. Fox, B.A. Popescu, Training in cardiac computed tomography: EACVI certification process, *Eur. Hear. J. - Cardiovasc. Imaging* 19 (2) (2017) 123–126.
- [34] A. Updegrove, N.M. Wilson, J. Merkow, H. Lan, A.L. Marsden, S.C. Shadden, SimVascular: An open source pipeline for cardiovascular simulation, *Ann. Biomed. Eng.* 19 (2) (2017) 525–541.
- [35] P. Sharma, L. Itu, X. Zheng, A. Kamen, D. Bernhard, C. Suci, D. Comaniciu, A framework for personalization of coronary flow computations during rest and hyperemia, in: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE*, 2012, pp. 6665–6668.
- [36] D.C. Flemister, H. Hatoum, V. Guhan, B. Zebhi, J. Lincoln, L.P. Crestanello, Effect of left and right coronary flow waveforms on aortic sinus hemodynamics and leaflet shear stress: Correlation with calcification locations, *Ann. Biomed. Eng.* 48 (12) (2020) 2796–2808.
- [37] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, S. Sridhar, Neural fields in visual computing and beyond, *Comput. Graph. Forum* 41 (2) (2022) 641–676.
- [38] A. Grop, L. Yariv, N. Haim, M. Atzmon, Y. Lipman, Implicit geometric regularization for learning shapes, in: *Proceedings of the 37th International Conference on Machine Learning, ICML '20*, JMLR.org, 2020, pp. 0–11.
- [39] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [40] J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: Learning continuous signed distance functions for shape representation, in: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, IEEE Computer Society, Los Alamitos, CA, USA, 2019, pp. 165–174.
- [41] J. Hagnberger, M. Kalimuthu, D. Muekamp, M. Niepert, Vectorized conditional neural fields: A framework for solving time-dependent parametric partial differential equations, in: *Forty-First International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, OpenReview.net, 2024, pp. 0–35.
- [42] P. Jin, S. Meng, L. Lu, MIONet: Learning multiple-input operators via tensor product, *SIAM J. Sci. Comput.* 44 (6) (2022) A3490–A3514.
- [43] J. Suk, B. Imre, J.M. Wolterink, LaB-GATr: geometric algebra transformers for large biomedical surface and volume meshes, 2024, *ArXiv abs/2403.07536*.
- [44] M.N. Rabe, C. Staats, Self-attention does not need  $O(n^2)$  memory, 2021, pp. 1–8, N/a.
- [45] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F.A. Hamprecht, Y. Bengio, A.C. Courville, On the spectral bias of neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019*, Long Beach, California, USA, in: *Proceedings of Machine Learning Research*, vol. 97, PMLR, 2019, pp. 5301–5310.
- [46] K. Crane, C. Weischedel, M. Wardetzky, The heat method for distance computation, *Commun. ACM* 60 (11) (2017) 90–99.
- [47] J. Suk, D. Alblas, B.A. Hutten, A. Wiegman, C. Brune, P. van Ooij, J.M. Wolterink, Physics-informed graph neural networks for flow field estimation in carotid arteries, 2024, pp. 1–10, N/a.
- [48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance deep learning library, in: *Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc.*, Red Hook, NY, USA, 2019, pp. 0–12.
- [49] M. Fey, J.E. Lenssen, Fast graph representation learning with PyTorch Geometric, in: *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019, pp. 0–9.
- [50] A. Hoogendoorn, A.M. Kok, E.M.J. Hartman, G. de Nisco, L. Casadonte, C. Chiastra, A. Coenen, S.-A. Korteland, K. Van der Heiden, F.J.H. Gijzen, D.J. Duncker, A.F.W. van der Steen, J.J. Wentzel, Multidirectional wall shear stress promotes advanced coronary plaque development: comparing five shear stress metrics, *Cardiovasc. Res.* 116 (6) (2019) 1136–1146.