Computer Methods in Applied Mechanics and Engineering 430 (2024) 117193

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

SECRET: Statistical Emulation for Computational Reverse Engineering and Translation with applications in healthcare

L. Mihaela Paun [a,*], Mitchel J. Colebank [b], Alyssa Taylor-LaPole [c], Mette S. Olufsen [c], William Ryan [a], Iain Murray [d], James M. Salter [e], Victor Applebaum [e], Michael Dunne [e], Jake Hollins [f], Louise Kimpton [e], Victoria Volodina [e], Xiaoyu Xiong [e], Dirk Husmeier [a]

[a] School of Mathematics and Statistics, University of Glasgow, Glasgow, G12 8SQ, UK
[b] Edwards Lifesciences Foundation Cardiovascular Innovation and Research Center, University of California, Irvine, CA, USA
[c] Department of Mathematics, North Carolina State University, Raleigh, NC, USA
[d] School of Informatics, University of Edinburgh, UK
[e] Department of Mathematics and Statistics, University of Exeter, Exeter, UK
[f] Department of Computer Science, University of Exeter, Exeter, UK

ARTICLE INFO

Dataset link: https://github.com/LMihaelaPaun/SECRET.git

Keywords:
Statistical emulation
Parameter inference
Uncertainty quantification
Computational fluid-dynamics
Personalised healthcare

ABSTRACT

There have been impressive advances in the physical and mathematical modelling of complex physiological systems in the last few decades, with the potential to revolutionise personalised healthcare with patient-specific evidence-based diagnosis, risk assessment and treatment decision support using digital twins. However, practical progress and genuine clinical impact hinge on successful model calibration, parameter estimation and uncertainty quantification, which calls for novel innovative adaptions and methodological extensions of contemporary state-of-the-art inference techniques from Statistics and Machine Learning. In the present study, we focus on two computational fluid-dynamics (CFD) models of the blood systemic and pulmonary circulation. We discuss state-of-the-art emulation techniques based on deep learning and Gaussian processes, which are coupled with established inference techniques based on greedy optimisation, simulated annealing, Markov Chain Monte Carlo, History Matching and rejection sampling for computationally fast inference of unknown parameters of the CFD models from blood flow and pressure data. The inference task was set as a competitive challenge which the participants had to conduct within a limited time frame representative of clinical requirements. The performance of the methods was assessed independently and objectively by the challenge organisers, based on a ground truth that was unknown to the method developers. Our results indicate that for the systemic challenge, in which an idealised case of noise-free data was considered, the relative deviation from the ground-truth in parameter space ranges from 10^{-5}% (highest-performing method) to 3% (lowest-performing method). For the pulmonary challenge, for which noisy data was generated, the performance ranges from 0.9% to 7% deviation for the parameter posterior mean, and from 35% to 570% deviation for the parameter posterior variance.

* Corresponding author.
E-mail address: Mihaela.Paun@glasgow.ac.uk (L.M. Paun).

https://doi.org/10.1016/j.cma.2024.117193
Received 6 May 2024; Received in revised form 20 June 2024; Accepted 21 June 2024
Available online 15 July 2024

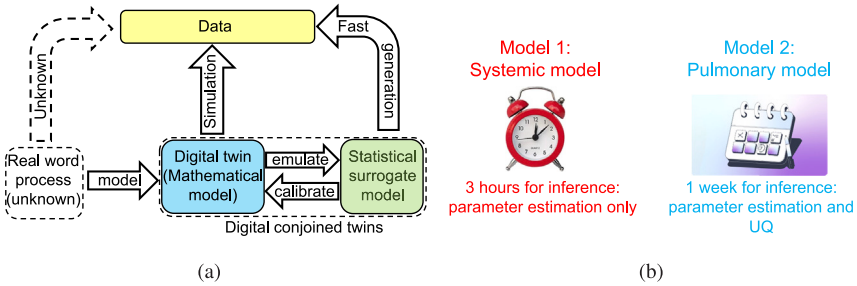


Fig. 1. Panel (a): Illustration of the concept of emulation. See main text for details. Panel (b): Overview of the two competition challenges: systemic model – parameter estimation, pulmonary model – parameter estimation and uncertainty quantification (UQ), with the associated time window within which submissions had to be made.

1. Introduction

There have been impressive advances in the physical and mathematical modelling of complex systems in the last few decades, which may for example be based on coupled partial differential equations (PDEs), and increasingly cover areas that until recently have been regarded as elusive for the quantitative sciences [1,2]. This includes complex ecosystems, e.g. modelling mitigation strategies for credible net zero implementations [3]; epidemiology, e.g. modelling the spread of pathogens and their infection patterns [4]; human physiology, e.g. assessing treatment effects for cardiovascular diseases [5]; urban studies, e.g. predicting traffic flow to ultimately prevent congestion [6]; and energy, e.g. developing forecasting and decision-support tools that support the energy sector to plan energy use and generation, and manage energy networks [7]. Common to all these examples is that the underlying mathematical equations describing the processes are intractable, that is, they have no closed-form solution. However, recent advancements in numerical methods and computer technology enable accurate and efficient numerical simulations, allowing the creation of a "digital twin", i.e., the virtual version of a complex real-world object or process that serves as a digital counterpart. Our particular focus is on cardiovascular modelling, which has the potential to revolutionise personalised healthcare with accurate patient-specific risk prediction and evidence-based treatment through digital replicas [8,9]. More specifically, our work focuses on cardiovascular modelling based on coupled nonlinear PDEs [10].

A PDE model of a complex physical system depends on various physical parameters, as well as initial and boundary conditions. Model parameters that cannot be measured or derived from first principles have to be estimated indirectly from available data. For instance, soft-tissue mechanical parameters of the heart muscle in vivo, related to the flexibility of the muscle fibres, have to be estimated from non-invasive magnetic resonance image scans of the heart [11]. This process, which is called model calibration or inference, is critical. If the model parameters are not estimated correctly, the digital twin provides a distorted virtual representation of the real-world object or process, and predictions made with it can be dangerously misleading [12]. Moreover, if the intrinsic uncertainty of the parameter estimation is not quantified accurately, any risk assessment may be flawed, leading to wrong decisions with potentially serious consequences, particularly in safety-critical applications.

A fundamental challenge for research and application is the fact that established parameter inference and uncertainty quantification (UQ) techniques assume that the inverse modelling (estimating the model parameters that generated a given data set) is intractable, whereas the forward modelling (getting data from a model with known parameters) is tractable. To elaborate on this, a linear model is double tractable, in that both the forward problem (which is just a linear weighted sum) and the inverse problem (the maximum likelihood solution, which is a product of three matrices and a data vector) have a closed-form solution. For more complex models, like neural networks (NNs), the inverse problem is intractable, in that there is no closed-form solution for the maximum likelihood solution and the likelihood surface is typically multimodal with many local optima, calling for the application of iterative numerical optimisation or sampling routines. The forward problem, however, has a closed-from solution, given by the definition of the NN itself, which is a nested structure of weighted summations subjected to nonlinear transformations. The practical solution usually requires numerical computation, but on a state-of-the-art computer that only takes the fraction of a second.

On the contrary, a complex physical model based on a nonlinear PDE system subject to various boundary conditions is not tractable: not only do we lack a closed-form solution of the equations themselves, we may not even be able to prove that such a solution exists. State-of-the-art approaches are based on the application of numerical routines using finite element discretisation. However, as opposed to the numerical computation required for getting an output from a NN, this typically leads to substantial computational costs in the order of several minutes or even hours, even when using advanced high-performance computing. We are thus facing a double intractable problem for which established iterative inference routines, which are based on repeatedly getting outputs from the model for different parameter configurations typically thousands of times, are practically infeasible.

We address this problem with emulation [13], a concept which is illustrated in Fig. 1(a). The objective is to develop a computationally tractable statistical surrogate model of the original intractable mathematical or physical model (the "digital twin"). This can be regarded as developing a digital conjoined twin that combines the domains of (i) the mathematical or physical model and (ii) statistics, machine learning, and data science providing a novel physical and mathematical sciences powerhouse. Data in the real world are produced by processes (e.g. physical, physiological, ecological, or socio-economic) that are unknown. The aim

of complex mathematical models is to build a digital twin of these processes that can give insight into the mechanisms driving them, predict outcomes and simulate data. The mathematical model can in principle be calibrated to minimise the discrepancy between simulated and real data. However, due to the high computational costs of repeated simulations, this approach is not viable in practice. We therefore build a statistical surrogate model to emulate the intractable mathematical model. This approach can be regarded as the creation of a pair of digital conjoined twins, whereby the statistical surrogate model is informed by the intractable mathematical model (the digital twin) for emulation, which in turn depends on the statistical surrogate model for calibration.

The current study considers two computational fluid-dynamics (CFD) models, for which statistical surrogate models are constructed. The surrogates are needed to enable fast statistical inference, which is essential for clinical translation. The first CFD model simulates haemodynamics in the large arteries of the systemic circulation within a patient with a single ventricle (Fontan) circulation. This specialised circuit is obtained after completion of three surgeries redirecting the vessels generating an effective single ventricle pump. Data used in this study are from a double outlet right ventricle (DORV) patient, a patient for which by birth the aorta is branching from the right ventricle [14]. The systemic model was developed to study how remodelling of the aorta impacts perfusion, comparing haemodynamics from DORV patients (the control group) with patients with hypoplastic left heart syndrome (HLHS) [15]. The focus was on predicting flow to the liver, which was an organ outside the imaged region. The latter is important as single ventricle Fontan patients experience liver disease [16,17]. The second application uses a similar mathematical model to simulate haemodynamics in the pulmonary arteries and veins developed for the analysis of pulmonary hypertension [10]. The aim is to determine how changes in the pulmonary arterial, venous, and microcirculatory tree affect the development of pulmonary hypertension [10]. For both models, understanding under what conditions disease type can be alleviated is of importance for improving treatment. Both models integrate imaging and haemodynamic (blood pressure and flow) data and provide insight into haemodynamics waveforms over a single cardiac cycle.

Before being used for prediction, these models need to be calibrated to data. The calibration process consists of inferring the unknown model parameters, which describe the material properties of the blood vessels (e.g., stiffness) and the geometric features of the distal vasculature, which is quantified through the "structured tree" model [18] from limited, noisy blood pressure and flow data.

The current study is focused on employing computational tools based on emulation for accurate, robust and computationally efficient inference of unknown model parameters from blood flow and pressure data. We discuss three state-of-the-art emulation approaches based on Gaussian Processes (GPs) [19,20] and deep residual NNs [21], which we couple with established statistical inference techniques based on greedy optimisation [22], simulated annealing [23] and UQ using Markov Chain Monte Carlo (MCMC) [24], and History Matching [25] with rejection sampling [26]. The performance of the proposed methods is comparatively evaluated with respect to estimation accuracy and UQ on the systemic and pulmonary models. An essential part of the assessment process is that the parameter estimation and UQ analysis must be accurate and conducted within a limited time interval. This limitation is with clinical translation in mind, to mimic clinical practice and decision support.

This study was run as part of a competition evaluating emulation methods, and the evaluation was conducted objectively in a blind process by the competition organisers. That is, participants were free to choose the methods of their choice, and we describe in Section 6 how we assessed their performance. Our article describes the three best-performing methods in Section 5 (with an overview shown in Table 3). We believe that our study is highly topical in that these methods are indicative of the current state-of-the-art, cover three principled methodological paradigms and, most importantly, describe innovative ways of how these methods have to be adapted and tuned in practice to meet clinical time constraints.

2. Physical models

The computational models consist of three parts: (1) a network forming the domain in which haemodynamics are predicted, (2) a system of PDEs that simulate haemodynamics in the large, proximal vessels and (3) boundary conditions specifying the inflow and form a representation of the microvasculature. The two models are distinct in their physiological representation and vascular components, but are derived from the same theoretical foundations.

2.1. Computational domain

For both applications the computational domain (a labelled tree) representing the vascular geometry is generated from medical images. Further details about the computational domain can be found in Section 1.1 of the Supplement.

2.1.1. Systemic model

The systemic model, shown in Fig. 2(a), predicts haemodynamics (blood flow and pressure) in the systemic arterial tree. The patient studied here has a double outlet right ventricle. To gain insight into the patient's physiology, this study models haemodynamics in a network including seven vessels from the aortic trunk and the proximal head and neck vessels (shown in Fig. 2(a)).

2.1.2. Pulmonary model

The pulmonary model, shown in Fig. 2(b) predicts haemodynamics (blood flow and pressure) in the pulmonary arteries and veins under normotensive conditions (denoting normal blood pressure). The model domain (shown in Fig. 2(b)) includes fifteen proximal, large arteries that are connected to twelve proximal, large veins.

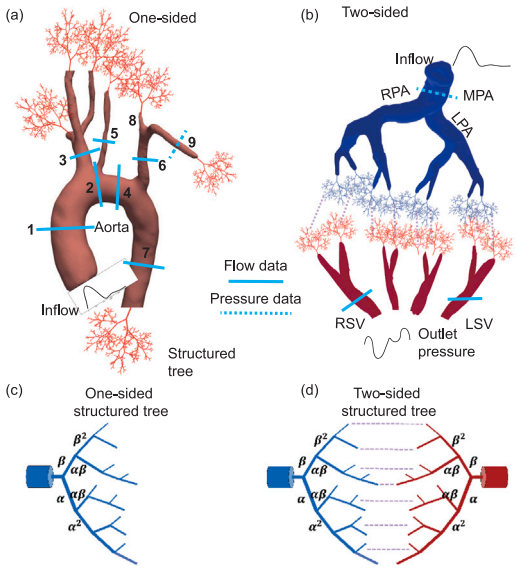


Fig. 2. Segmented and rendered 3D surfaces for (a) the systemic and (b) the pulmonary model. For both models data are measured at locations marked with continuous (flow data) and dashed (pressure data) lines. For both models the small vessels are represented by structured trees. The systemic model uses a one-sided tree (c), while the pulmonary model uses a two-sided tree (d). For both models, the structured tree is parameterised by radii scaling factors α and β, a length-to-radius ratio, lrr (two length-to-radius ratios in the pulmonary model), and a minimum radius, $r_{\min}$, describing the radius where the structured tree terminates.

2.2. Large vessel fluid dynamics

In the large vessels, blood flow $q(x,t)$ (cm^3/s), pressure $p(x,t)$ mmHg, and vessel area $A(x,t)$ (cm^2) are computed by solving a 1D fluid dynamics model. The blood is assumed to be incompressible, Newtonian, and viscous with constant density, $\rho = 1.057$ (g/cm^3), and viscosity, $\nu = 0.032$ (g/cm/s). The vessels are assumed to be cylindrical, axisymmetric, and impermeable with a circular cross-section. Pressure, flow, and area satisfy conservation of mass and momentum balance equations of the form [18]

$$\frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} = 0, \quad \frac{\partial q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q^2}{A}\right) + \frac{A}{\rho}\frac{\partial p}{\partial x} = -\frac{2\pi\nu R}{\delta}\frac{q}{A}, \tag{1}$$

where $0 \le x \le L$ are the axial coordinates and $0 \le t \le T$ are the temporal coordinates, and $\nu = \frac{\mu}{\rho}$ (cm^2/s) is the kinematic viscosity. $R(x,t)$ is the radius of the vessel, $\delta = \sqrt{\nu T/2\pi}$ (cm) is the boundary layer thickness, $T(s)$ is the cardiac cycle duration. To model the stress–strain behaviour of the arterial wall, two common, linear stress–strain relationships are considered

$$p(x,t) - p_0 = \frac{4}{3}\frac{Eh}{r_0}\left(\sqrt{\frac{A}{A_0}} - 1\right), \quad p(x,t) - p_0 = \frac{4}{3}\frac{Eh}{r_0}\left(1 - \sqrt{\frac{A_0}{A}}\right), \tag{2}$$

where E (g/cm/s^2) is Young's modulus, h (cm) is the vessel wall thickness, p_0 (g/cm/s^2) is the reference pressure, r_0 (cm) is the reference radius, and A_0 (cm^2) is the reference area. The first pressure–area relationship is used in the systemic model, while the second formulation is used in the pulmonary model. Both wall models have been analysed extensively in the systemic circulation [27], and each have their own benefits and limitations in describing in-vivo haemodynamics. We consider both stress–strain relationships for diversification purposes in the competition.

Boundary conditions are specified at the inlet and outlet of each vessel. For both models, a flow waveform is imposed from magnetic resonance imaging (MRI)-measured data at the network inlet. The pulmonary model requires an additional distal boundary condition, for which we use a dynamic, left-atrial pressure waveform. At the vessel junctions mass conservation and pressure continuity are enforced via

$$q_p(L,t) = q_{d_1}(0,t) + q_{d_2}(0,t), \quad p_p(L,t) = p_{d_1}(0,t) = p_{d_2}(0,t), \tag{3}$$

where subscript p denotes the parent vessel and subscripts d_1 and d_2 denote the daughter vessels. As outflow boundary conditions at the terminal vessels, we use structured tree boundary conditions as they represent small blood vessels down to the capillary level, which cannot be obtained from micro-computed tomography (CT) due to insufficient image resolution. Section 1.2 of the Supplement contains additional details about the fluid dynamics in the large vessels.

Table 1
Range of values for the biophysical parameters of the systemic model and the "ground-truth" parameter values that generated the competition noise-free data.

Parameter	f_2^{LA}	f_3^{LA}	f_2^{MV}	f_3^{MV}	α
Range	$[-45,-25]$	$[2\cdot10^5, 9\cdot10^5]$	$[-45,-25]$	$[2\cdot10^5, 9\cdot10^5]$	$[0.85, 0.94]$
"Ground-truth" value	-32.9	$4.26\cdot10^5$	-40.6	$6.43\cdot10^5$	0.88

2.3. Small vessel fluid dynamics

The core of the small vessel model is the structured tree shown in Fig. 2(c) and described in detail in [18]. This self-similar bifurcating network is composed of straight vessels in which all properties relate to the vessel radii. The radius of the daughter vessels scale with factors α and β from the parent vessel, and the length of each vessel is related to its radius by a constant factor lrr. A minimum radius, r_{min} is imposed, which describes the radius where the structured tree terminates.

The structured tree differs for the two models. For the systemic model, we only include arterial vessels that branch until small arterioles (shown in Fig. 2(c)). In the pulmonary model, we use a two-sided structured tree (shown in Fig. 2(d)) allowing blood to be transmitted between the arterial and venous networks. Further details about the one-sided and two-sided structured trees can be found in Section 1.3 of the Supplement.

2.4. Numerical methods

The model equations are non-dimensionalised and solved using the two-step Lax–Wendroff method [28]. One simulation takes approximately one minute for the systemic model and two minutes for the pulmonary model on our hardware (RedHat Enterprise Linux 6 machine with Intel(R) Xeon(R) CPU E5-2680 v2 2.80 GHz and 32 GB RAM).

2.5. Model parameters

The fluid properties (ρ, ν, δ) are assumed known and constant, see Section 2.2 and Table 1 in the Supplement for details.

Vessel stiffness: Vessel stiffness in the large arteries, micro-vasculature (small vessels) and large veins (the latter for the pulmonary model only) is modelled using the following expression

$$\frac{Eh}{r_0} = f_1 \exp(f_2 r_0) + f_3, \tag{4}$$

where we define f_1^{LA} (g/cm/s^2), f_2^{LA} (cm^{-1}), f_3^{LA} (g/cm/s^2) as the stiffness parameters of the large arteries, $f_1^{\mathrm{MV}}, f_2^{\mathrm{MV}}, f_3^{\mathrm{MV}}$ are the stiffness parameters of the micro-vasculature, and $f_1^{\mathrm{LV}}, f_2^{\mathrm{LV}}, f_3^{\mathrm{LV}}$ are the stiffness parameters of the large veins.

Structured tree parameters: The model also has structured tree parameters: α and β, which are constants that define the asymmetry of the structured tree, the length to radius ratio, lrr (one arterial lrr for the one-sided structured tree of the systemic model, and one arterial and one venous lrr for the 2-sided structured tree of the pulmonary model), and the minimum radius (r_{min}).

Thus, the full set of parameters that was initially considered for inference is as follows:

- Systemic model: $\boldsymbol{\theta}_{\mathrm{full}} = \{f_1^{\mathrm{LA}}, f_2^{\mathrm{LA}}, f_3^{\mathrm{LA}}, f_1^{\mathrm{MV}}, f_2^{\mathrm{MV}}, f_3^{\mathrm{MV}}, \alpha, \beta, lrr_{\mathrm{A}}, r_{\mathrm{min}}\}$.
- Pulmonary model: $\boldsymbol{\theta}_{\mathrm{full}} = \{f_1^{\mathrm{LA}}, f_2^{\mathrm{LA}}, f_3^{\mathrm{LA}}, f_1^{\mathrm{MV}}, f_2^{\mathrm{MV}}, f_3^{\mathrm{MV}}, f_1^{\mathrm{LV}}, f_2^{\mathrm{LV}}, f_3^{\mathrm{LV}}, \alpha, \beta, lrr_{\mathrm{A}}, lrr_{\mathrm{V}}, r_{\mathrm{min}}\}$.

Local sensitivity and identifiability analyses performed around the competition parameter values revealed a lack of identifiability between the parameters. For the sake of the competition, that is, to allow the method assessment in both functional *and* parameter space, we only consider for inference a subset of the parameters that are identifiable and most influential on the model outputs. In a clinical application we would address identifiability problems by the integration of physiological prior knowledge; see [11], in particular the discussion around Fig. 6. However, this is beyond the remit of the present article, whose focus is on evaluating emulation and inference performance. In Section 1.5 of the Supplement we provide brief details of the methodology employed for the local sensitivity and identifiability analyses performed, and present the results obtained on the two CFD models. The final subset of inferable parameters, which are uniquely identifiable and most influential on the model outputs, are as follows:

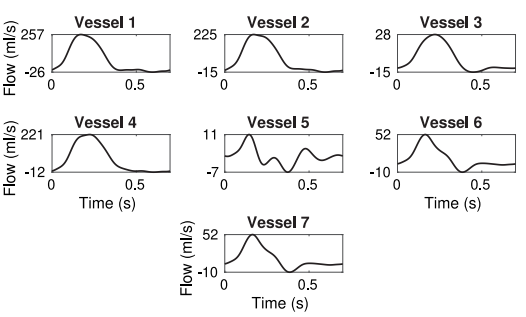
- Systemic model: $\boldsymbol{\theta}_{\mathrm{subset}} = \{f_2^{\mathrm{LA}}, f_3^{\mathrm{LA}}, f_2^{\mathrm{MV}}, f_3^{\mathrm{MV}}, \alpha\}$, with associated physiological ranges presented in Table 1.
- Pulmonary model: $\boldsymbol{\theta}_{\mathrm{subset}} = \{f_3^{\mathrm{MV}}, \alpha, lrr_{\mathrm{A}}, lrr_{\mathrm{V}}\}$, with associated physiological ranges given in Table 2. Details about the parameter ranges are given in Section 1.5 of the Supplement.

3. Competition data

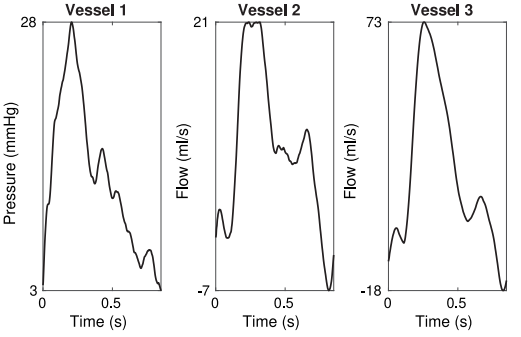
To assess the performance of the competing methods presented, the competition organisers use synthetic data, that is data simulated from the aforementioned CFD models with preset parameter values.

Table 2
Range of values for the biophysical and error model parameters of the pulmonary model and the "ground-truth" parameter values that generated the competition noisy data.

Parameter	f_3^{MV}	α	lrr_{A}	lrr_{V}	l	σ_m^2
Range	$[9 \cdot 10^4, 3 \cdot 10^5]$	[0.83,0.89]	[20,50]	[20,50]	[0, 0.85]	[0, 22.5]
"Ground-truth" value	$2.5 \cdot 10^5$	0.885	35	25	0.1	5.8



(a) Systemic noise-free data: flow in seven large arteries marked in Figure 2(a).



(b) Pulmonary noisy data: pressure in one large artery and flow in two large veins marked in Figure 2(b).

Fig. 3. Competition data for both systemic and pulmonary models.

3.1. Systemic model

For the systemic circulation model, the organisers have generated synthetic, *noise-free* flow time series from 7 blood vessels, indicated with continuous lines in Fig. 2(a) at one spatial point (the vessel midpoint) and the systolic and diastolic points in a pressure time series from one blood vessel, marked with a dashed line in Fig. 2(a). Competition data are reflective of actual patient data from [15]. The competition data have been generated using the parameter values for $f_2^{\mathrm{LA}}, f_3^{\mathrm{LA}}, f_2^{\mathrm{MV}}, f_3^{\mathrm{MV}}, \alpha$ shown in Table 1 and are illustrated in Fig. 3(a).

3.2. Pulmonary model

For the pulmonary circulation model, the organisers have generated synthetic, *noisy* pressure time series data in the main pulmonary artery, marked with a dashed line in Fig. 2(b), and flow time series in two veins, marked with continuous lines in Fig. 2(b) at one spatial point (the vessel midpoint). The competition data, shown in Fig. 3(b), mimic typical real, noisy data [29–31]. The data were generated using the parameter values for $f_3^{\mathrm{MV}}, \alpha, lrr_{\mathrm{A}}, lrr_{\mathrm{V}}$ shown in Table 2.

The noise added to the data is Gaussian, additive and correlated in time. To generate the noise, a GP [32] with a Matérn 3/2 kernel has been used. More specifically, a GP has been fitted to the residuals in time, as follows:

$$f(t)|\boldsymbol{\xi} \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}|\boldsymbol{\xi}) \approx \mathcal{GP}(\mathbf{0}, \mathbf{C})|\boldsymbol{\xi}), \tag{5}$$

where $\mathbf{K} = [k(t_i, t_j)]_{i,j=1}^m$ (indexed by time, t) is the $m \times m$ variance–covariance matrix of f, and $\mathbf{C} = \mathbf{K} + \sigma^2\mathbf{I}$ is used solely for numerical stabilisation purposes during the inversion of the covariance matrix, with $\sigma^2 = 10^{-6}$. Also, $\boldsymbol{\xi}$ contains the covariance function (kernel) hyperparameters, called error model parameters, which are assumed common to all vessels. It should be noted that the measurement devices for pressure and flow are distinct: pressure is measured invasively by a right heart catheter, while flow is typically obtained non-invasively using MRI. However, pressure catheters are more susceptible to the dynamic wall motion of the pulmonary arteries, introducing measurement noise. Similarly, MRI is a non-invasive imaging technique that requires averaging over multiple measurements at different heartbeats. In the absence of detailed insight into the relative uncertainties attributed to these data modalities, we reasonably assume that the noise variance is similar for both data sources.

The Matérn 3/2 kernel is defined as

$$k(r|\boldsymbol{\xi}) = \sigma_m^2(1 + \sqrt{3}r)\exp(-\sqrt{3}r), \quad r = \sqrt{\frac{(t_i - t_j)^2}{l^2}}, \tag{6}$$

where $\boldsymbol{\xi} = (\sigma_m^2, l)$, with σ_m^2 being the marginal variance of the function (amplitude), and l being the lengthscale of the input (time) variable.

For the competition, the error model parameter values shown in Table 2 were used for noise generation to ensure a signal-to-noise ratio between 10 and 100 for all 3 vessels.

4. Competition challenges

The organisers released the code for the two CFD models several months in advance of data release, to allow participants to familiarise themselves with running the code, and build emulators for the model outputs. At the time, the organisers gave preliminary indications of what the inference tasks would be, but the exact, model-specific competition challenges were released together with the data. In Sections 4.1 and 4.2 we present the competition challenges for each of the two CFD models and the time window within which submissions had to be made, see Fig. 1(b) for an overview.

4.1. Systemic model

The competition challenge for the systemic model was to estimate the set of five biophysical parameters that generated the competition data. The time limit allowed for the parameter estimation was *three hours*, which reflects a typical time window within which a diagnosis or treatment decision has to be made in a clinical A&E setting.

4.2. Pulmonary model

For the pulmonary model, the competition challenge was to provide parameter estimates for the four biophysical parameters and quantify the uncertainty of the estimation in *one week*. The time limit reflects a typical time frame within which a medical laboratory has to operate. This is not as stringent as for a clinical A&E setting, but still requires any quantitative analysis to be conducted in a timely manner.

Competition organisers invited participants to submit estimates of the biophysical parameters and a measure of estimation uncertainty, that is 500 samples from the posterior distribution of the biophysical parameters. The error parameters are nuisance parameters, meaning their estimation was not assessed directly in the competition, but may be relevant for the accurate inference of the biophysical parameters.

The participants were instructed to use a uniform prior with support within ranges given in Table 2. While the range for the biophysical parameters was chosen to be biologically meaningful, the organisers chose the range for the error parameters as follows. The kernel lengthscale takes the range of the time input, i.e. [0, 0.85] seconds. To find the range for the kernel amplitude, residuals were generated repeatedly (100 times) with preset values (found in Table 2), and the variance of each residual instantiation was computed. The upper bound of the range is given by the maximum variance value recorded to which 50% was added (obtaining 22.5), to avoid being overly-conservative. The lower bound of 0 imposes positivity in the amplitude prior.

5. Methodology

In this study, the participants have employed state-of-the-art, data-driven emulation approaches [33] based on Bayesian non-parametric models, more specifically GPs [19,20], as well as deep learning models, more specifically, residual NNs [21]. These emulation paradigms represent recent methodology widely adopted in the machine learning community, and have been used in our study to approximate features of the complex CFD models (“simulator”), with the aim to perform fast statistical inference of unknown simulator parameters. Thus, emulation has been coupled with established statistical inference approaches based on greedy optimisation [22], simulated annealing [23] and UQ using History Matching [25] coupled with rejection sampling [26], as well as MCMC [24].

In Table 3 we present an overview of the 3 methods employed in this study, where we define the method as the marriage between the emulation approach and the inference scheme utilised. The output from the simulators is multiple time series. To emulate this multivariate output, the developer of method I has first used Principal Component Analysis (PCA) for dimensionality reduction of every time series, and has subsequently emulated each principal component with independent GPs. The developer of method II has used PCA and has employed a deep residual NN to emulate the multivariate set of principal components. PCA has only been performed for the systemic model, which has a large output; for the pulmonary model, for which there is less output, every multivariate time series has been emulated. The developer of method III has emulated every time series using a multivariate output GP with a Kronecker product approximation for the covariance matrix [19].

The contestants have used the emulators thus created to perform parameter estimation and quantify the parameter uncertainty using History Matching coupled with rejection sampling (method I), simulated annealing and MCMC (method II), and local (greedy) optimisation with multiple restarts and MCMC (method III). A description of the emulation and inference approaches employed follows below.

5.1. Method I

5.1.1. GP emulation

Let p denote the number of inputs, and m the number of outputs. Given n design points $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n)$, sampled from a p-dimensional parameter space $\mathcal{X} \subset \mathbb{R}^p$, we simulate the true model $\mathbf{f}(\cdot)$ at these inputs for fixed time points, generating blood flow and pressure. We combine the model outputs at a single input into an m-dimensional vector $\mathbf{f}(\boldsymbol{\theta}, \mathbf{t}) = (f(\boldsymbol{\theta}, t_1), \dots, f(\boldsymbol{\theta}, t_m))^\mathsf{T}$, where the superscript T indicates transposition, giving an $m \times n$ matrix of model outputs, $\mathbf{F}(\boldsymbol{\Theta}, \mathbf{t}) = (\mathbf{f}(\boldsymbol{\theta}_1, \mathbf{t}); \dots; \mathbf{f}(\boldsymbol{\theta}_n, \mathbf{t}))$. As the size of the output, m, increases, for computational convenience, and to capture common patterns across the model output, GP emulation can

Table 3
Methods employed in this study for emulation of simulator outputs and inference of unknown simulator parameters.

	Emulation	Inference
Method I	Bayesian non-parametric (univariate GPs)	History Matching[1,2] & rejection sampling[2]
Method II	Deep learning (residual NN)	simulated annealing[1]/MCMC[2]
Method III	Bayesian non-parametric (multivariate GPs)	greedy optimisation[1]/MCMC[2]

Note: superscripts [1] and [2] refer to the systemic and pulmonary model, respectively.

be combined with dimension reduction, such as PCA [34–38], and is the approach used here. The model output is decomposed into a linear combination of basis vectors $\boldsymbol{\gamma}_i$, $\mathbf{f}(\boldsymbol{\theta},\mathbf{t}) = \boldsymbol{\mu}(\mathbf{t}) + \sum_{i=1}^{q} c_i(\boldsymbol{\theta},\mathbf{t})\boldsymbol{\gamma}_i + \boldsymbol{\epsilon}(\boldsymbol{\theta},\mathbf{t})$, where $\boldsymbol{\mu}(\mathbf{t})$ is the mean of the simulator runs $\mathbf{F}(\boldsymbol{\Theta},\mathbf{t})$; $c_i(\boldsymbol{\theta},\mathbf{t})$ are coefficients; basis $\boldsymbol{\Gamma}_q = (\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_q)$ is often taken from the singular value decomposition of $(\mathbf{F}(\boldsymbol{\Theta},\mathbf{t}) - \boldsymbol{\mu}(\mathbf{t}))^{\mathsf{T}}$; and $\boldsymbol{\epsilon}(\boldsymbol{\theta},\mathbf{t})$ represents the part of $\mathbf{f}(\boldsymbol{\theta},\mathbf{t})$ not explained by the basis. Given $\boldsymbol{\Gamma}_q$ and $\mathbf{F}(\boldsymbol{\Theta},\mathbf{t})$, the output at $\boldsymbol{\theta}$ is now described by q coefficients instead of the m original outputs, with $q \ll m$. Independent GP emulators [39] are then fitted for each set of coefficients. Further method details can be found in Section 2.1.1 of the Supplement.

5.1.2. Inference with History Matching

For parameter inference, the "best input approach" model [33] was adopted:

$$\mathbf{y}(t) = \mathbf{f}(\boldsymbol{\theta}^*, t) + \mathbf{e}(t) + \boldsymbol{\eta}(t), \tag{7}$$

where $\mathbf{y}(t)$ are observations (synthetic data) indexed by time, $\boldsymbol{t} = (t_1,\ldots,t_m)$, $\mathbf{f}(\boldsymbol{\theta}^*,\boldsymbol{t})$ is the computer model output (blood flow and pressure) at the "best input" $\boldsymbol{\theta}^*$, $\boldsymbol{\eta}(\boldsymbol{t})$ is the model discrepancy term, and $\mathbf{e}(\boldsymbol{t})$ is the observational error. History Matching can then be used to rule out regions of parameter space that are not consistent with observations using a distance metric (implausibility function) $\mathcal{I}(\boldsymbol{\theta})$ [40]. For m-dimensional observations $\mathbf{y}$, using the model from Eq. (7), we can write [25,37]:

$$\mathcal{I}(\boldsymbol{\theta}) = (\mathbf{y}(\mathbf{t}) - \mathrm{E}[\mathbf{f}(\boldsymbol{\theta},\mathbf{t})])^{\mathsf{T}}(\mathrm{Var}[\mathbf{f}(\boldsymbol{\theta},\mathbf{t})] + \boldsymbol{\Sigma}_{\mathbf{e}} + \boldsymbol{\Sigma}_{\boldsymbol{\eta}})^{-1}(\mathbf{y}(\mathbf{t}) - \mathrm{E}[\mathbf{f}(\boldsymbol{\theta},\mathbf{t})]), \tag{8}$$
$$\mathbf{e}(\mathbf{t}) \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{e}}),\ \boldsymbol{\eta}(\boldsymbol{t}) \sim \mathcal{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}}),$$

where $\mathbf{y} \in \mathbb{R}^m$, $\boldsymbol{\Sigma}_{\mathbf{e}}, \boldsymbol{\Sigma}_{\boldsymbol{\eta}} \in \mathbb{R}^{m\times m}$. "Not Ruled Out Yet" (NROY) space, the space of not implausible ($\neq$ plausible) points, is defined as: $\mathcal{X}_{NROY} = \{\boldsymbol{\theta} \in \mathcal{X} | \mathcal{I}(\boldsymbol{\theta}) < T\}$, with $\boldsymbol{\theta}$ only ruled out if, given error tolerances, it is unlikely (implausible) that $\mathbf{f}(\boldsymbol{\theta})$ is "close-enough" to $\mathbf{y}$, depending on the threshold value T chosen. Hence the term Not Ruled Out "Yet" indicates that we cannot say that for all the points in this space we expect the model output to be close to the observation, but that there is not yet enough evidence to rule them out. For instance, a high emulator uncertainty reduces the value of the implausibility function. Further method details are presented in Section 2.1.2 of the Supplement.

Systemic model: Prior to data being released, method I developer ran the model 1250 times using a Latin hypercube space-filling design to explore the use of different emulation and calibration techniques tested on proxy observations. As there was no noise in the observation, an arbitrary variance was included to represent tolerance to error. At wave 1, basis emulators were trained using $n = 200$ simulations of the model, and the implausibility was calculated across the remaining available simulations. Wave 2 refined this by sampling a new $n = 200$ design from the original 1250, conditional on minimising the implausibility at wave 1 to ensure a greater density of points closer to the data. Using these refined emulators, the contestant evaluated the expectation and variance at a Latin hypercube of 100,000 samples of $\boldsymbol{\theta}$, calculated the implausibility $\mathcal{I}(\boldsymbol{\theta})$ for these points, and selected at wave 3 the $n = 10$ points that minimised $\mathcal{I}(\boldsymbol{\theta})$ across this sample. At wave 4, the participant obtained $n = 10$ simulation runs by sampling around the "best" input from wave 3, which was the parameter estimate submitted for the competition. More details are offered in Section 2.1.3 of the Supplement.

Pulmonary model: The participant used the same basis emulator structure and general iterative procedure as for the systemic model (see Algorithm 1 in the Supplement), with the change that the implausibility was instead calculated across combinations of (θ, l) (due to the unknown, varying correlation structure) with $\sigma_m^2 = 22.5$ (providing an upper bound on $\mathcal{I}(\theta, l)$ across the prior for σ^2). Training the emulator using a 100-member Latin hypercube design ruled out all $l \geq 0.2$, but with the large variance, no settings of θ could be ruled out across all l. Two new waves were performed by sampling inputs with relatively low implausibility (runs either expected to be close to $\mathbf{y}$, or those with high emulator variance), resulting in a greater density of model simulations more consistent with the truth, and an emulator was trained on this set of 300 simulations. The contestant calculated the likelihood for combinations of (θ, l, σ_m^2) (for combinations of (θ, l) in NROY, and for uniformly sampled σ_m^2) using the wave 3 emulator, and performed rejection sampling over these to obtain 500 samples of θ, which were submitted for the competition. The posteriors contain the truth, with the larger uncertainty relative to the true model due to emulator uncertainty $\mathrm{Var}[f(\theta)]$ being non-zero. After the results were released, applying History Matching with the known, fixed error structure, gave similarly wide posteriors, with additional waves expected to reduce the uncertainty. More details are offered in Section 2.1.4 of the Supplement.

5.2. Method II

Method II developer exploited parallel computation, and generated 10^5 simulations from the CFD models using 64 core machines in less than a week; jobs were run using GNU Parallel [41]. The parameter samples were generated uniformly at random inside a box given by the lower and upper parameter bounds. The best training and validation performance was from deep residual NNs. Preliminary experiments with xgboost [42] and linear regression with random basis functions [43] were less successful.

5.2.1. NN emulation

Architecture: A deep fully-connected feedforward NN was used, with "PReLU" activations [44] and 15 residual layers [21]. In detail, the network first transforms the input parameters $\boldsymbol{\theta}$ into a new 'hidden layer' representation, $\mathbf{h}^{(0)} = g(\mathbf{W}^{(0)}\boldsymbol{\theta} + \mathbf{b}^{(0)})$, where in this section weight matrices $\mathbf{W}$ and vectors of biases $\mathbf{b}$ are free parameters, and g is a piecewise-linear "PReLU" activation. The net then repeatedly retransforms those values using L *residual layers*, $\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + g(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}), \ell = 1 \dots L$. The residual connection ($\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \cdots$) makes training many layers work in practice, but restricts all of the hidden vectors to be the same length H. The first weight matrix $\mathbf{W}^{(0)}$ transforms the representation up to that dimensionality. The residual layer weight matrices are then all $H \times H$. Finally the simulator output is predicted from the final hidden layer, $\mathbf{f} = \mathbf{W}^{(\text{out})}\mathbf{h}^{(L)} + \mathbf{b}^{(\text{out})}$. The weight matrix $\mathbf{W}^{(\text{out})}$ transforms the final hidden representation to the length of all of the simulator outputs that we are using, concatenated together.

In theory, we do not need a deep NN for a flexible function: we could use no residual layers ($L = 0$), and set H large to have a single wide hidden layer, similar to transforming the parameters with random basis functions and performing linear regression on top, but with learned basis functions. However, empirically better fits were obtained for the same compute cost with deep narrower networks. H was set to 100 and 128 for the pulmonary and systemic challenge, respectively, and $L = 15$. These choices were not carefully tuned, but gave predictions on held out parameters that, by eye, lay on top of the simulator's output.

NN fitting: The network was implemented in Equinox [45], using its default settings to initialise the parameters. Not knowing the tasks ahead of time, the participant simply minimised the sum of square errors between the true simulated output $\mathbf{y}(\boldsymbol{\theta})$ and the NN's output $\mathbf{f}(\boldsymbol{\theta})$, averaged over training set examples. The tuning-free learned optimiser VeLO [46] was used, to avoid tuning the learning rate in the Adam algorithm [47]. The contestant iterated over a training set of 8×10^4 examples for 1000 epochs, in batches of size 100. Training and validation errors were usually quite similar, so no form of regularisation such as weight decay was used. VeLO has substantial overhead for small networks with small batch sizes, and so training took a few hours on a GPU, which was still far quicker than obtaining the simulations.

Reducing training cost with PCA: The participant used one trick to reduce the cost for the systemic challenge, with its larger simulation output. Given a training set of simulation outputs, each simulation in the training set has a final hidden layer vector $\mathbf{h}^{(L)}$, which is linearly transformed into a prediction of the simulation output. If the linear layer parameters, and all of the hidden layer vectors could be set arbitrarily, the best square error is obtained by a PCA [48] fit of the simulation outputs. The participant therefore replaced the targets of the NN with 128-dimensional linear transformations of the simulation outputs, provided by PCA, and later reconstructed any NN predictions into full simulation outputs using the same PCA representation fitted to the training set.

5.2.2. Inference with simulated annealing/MCMC

Systemic model: For the systemic challenge, the participant attempted to optimise the parameters by minimising the square error between the simulation output and the competition data. To produce the results submitted for the competition, the NN surrogate was used instead of the true simulator. The contestant adopted a simulated-annealing-like heuristic [23], by running MCMC [24] using the `emcee` ensemble sampling package [49], which usually works well on low-dimensional posteriors without tuning, and is able to exploit the ability to evaluate the NN for many different parameters in parallel. Emcee was run with 200 walkers for 1000 steps, with a spherical Gaussian observation model, starting at observation variance 500, which was annealed down by a factor of 0.99 at each step. The emcee walkers were reinitialised around the current best point every 20 steps. Details of alternative inference approaches employed are presented in Section 2.2 of the Supplement.

Pulmonary model: For the pulmonary challenge the contestant attempted to sample plausible parameters given the noisy observation using MCMC on the joint posterior distribution of the 4 unknown simulator parameters and the 2 hyperparameters of the GP in the observation model. To evaluate the likelihood of the parameters, the GP model provided by the organisers was assumed, with the mean observation set by the NN surrogate rather than the original simulator. Given the posterior is only 6 dimensions, emcee was used. The Markov chain appeared to mix quickly, with emcee reporting mixing (measured by max auto-correlation time) in around 50 steps. 128 walkers could be run for 500 steps in less than 10 min on a laptop from 2016. To reduce the chance of MCMC error, the contestant ran 500 walkers for 5000 steps, discarding 500 as burn-in.

5.2.3. Post-competition method improvements

Large Network: After the competition the participant also fitted an even larger NN for the pulmonary challenge. The network architecture and training data was exactly the same as before, except each residual layer had 1000 units instead of 100. In addition the loss function combined square error, with the likelihood of the Matérn GP observation model used by the competition organisers. This loss function was designed to make errors in the surrogate have a small effect on likelihood computations within MCMC. Training with VeLO on a GPU took less than 6 h. The parameter inference scheme was the same as for the competition.

5.3. Method III

5.3.1. GP emulation

GPs [32] were utilised to emulate the time series output of both the pulmonary and systemic models. For either model, the simulator output for input vector of physiological parameters $\boldsymbol{\theta}$, given by $\mathbf{f}(\boldsymbol{\theta}, t) = \mathbf{y}(t) = (y(t_1), y(t_2), \dots y(t_m))$ is a *multivariate* time series composed of $m = 512$ points. To avoid implementing an onerous multioutput GP for the multivariate output function, time may be introduced as an input variable alongside $\boldsymbol{\theta}$ [19], in which case a new 1-to-1 simulator function may be defined, $f(\boldsymbol{\theta}, t) = y(t)$, which defines a *univariate* output. Therefore, using a GP as a surrogate model of $f(\boldsymbol{\theta}, t)$ allowed for emulating a single output function. Further, separability in the kernel can be imposed between inputs $\boldsymbol{\theta}$ and t [19], $k((t_i, \boldsymbol{\theta}_i), (t_j, \boldsymbol{\theta}_j)) = k_t(t_i, t_j)k_{\boldsymbol{\theta}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$,

which allows representing the joint (full) covariance matrix as the Kronecker product between two smaller matrices, $\mathbf{K}(\boldsymbol{\Theta}_{\theta,t}, \boldsymbol{\Theta}_{\theta,t}) = \mathbf{K}_t(\mathbf{t},\mathbf{t}) \otimes \mathbf{K}_{\boldsymbol{\theta}}(\boldsymbol{\Theta},\boldsymbol{\Theta})$, with the aim to reduce computational complexity of the covariance matrix inversion. Further details are presented in Section 2.3 of the Supplement.

Space-filling designs of $n = 10{,}000$ and 15,000 points, obtained via Latin hypercube sampling, were used to build the GPs for the pulmonary and systemic models, respectively. The kernel hyperparameters were optimised using an adaptive Nelder–Mead algorithm [50], using a subset of 2000 training points. Instead of using the entire training dataset in the predictive equations for a test point $\boldsymbol{\theta}^*$, a local subset of 200 points nearest to $\boldsymbol{\theta}^*$ in standardised Euclidean space was used. Further implementation details can be found in Section 2.3 of the Supplement.

5.3.2. Inference with optimisation/MCMC

Systemic model: For the systemic model, the participant attempted to recover the parameters which produced the noise-free data by minimising a root mean squared error loss function using a greedy optimisation method, i.e. the Powell method [22]. 10 searches were run in parallel and once a minimum was obtained, the minimising parameters and corresponding simulator output were added to the emulator's dataset before restarting the search. For the first two hours out of three, points were uniformly sampled from the parameters' viable ranges, before restricting the sampling bounds to the region the optimum was believed to lie in based on the results of the first two hours.

Pulmonary model: For inference in the pulmonary model, the participant used an Adaptive Metropolis (AM) sampler [51]. The simulator output was replaced by the GP predictive mean for both flow and pressure in the likelihood function. Before collecting a final sample, an exploratory phase was carried out where the emulator was improved by allowing the MCMC algorithm to explore the high density regions of the posterior and adding every 100^{th} set of parameter values and corresponding simulator output to the GP's dataset, for a total of 100 additional simulator runs. 100,000 points were sampled from the posterior and thinned down to 500 to submit for the competition.

5.3.3. Post-competition method improvements

Improved GP hyperparameter search: After the competition, the GP models were refit using TensorFlow, which made hyperparameter optimisation using the entire sets of training points for both models possible in a realistic timeframe. Using TensorFlow's GPU acceleration allowed for much faster likelihood evaluations and gradient-based optimisation via automatic differentiation. The Adam optimiser [47] was used to fit all GP hyperparameters. Predictions also made use of the entire training dataset, in contrast to the local GP approach used during the competition.

Improved inference scheme: For the systemic challenge, post-competition, the participant used the gradient-based optimiser Adam using the improved GP fits for parameter inference. Post-competition inference for the pulmonary model used the improved GP fits, and the inference process was similar to that for the competition, with the key difference of performing the exploratory phase on a tempered likelihood. The aim was to smoothen local modes of the target posterior introduced by emulation inaccuracy and hence allow the exploratory chain to explore the tail ends of the target posterior and include these points in the GPs' datasets. Further implementation details can be found in Section 2.3.1 of the Supplement.

6. Assessment criteria

In this section we proceed to describe the approaches used to evaluate the competition submissions.

6.1. Systemic model - accuracy

Given the competition data for the systemic model were noise-free, the assessment criteria evaluate accuracy of estimates only (no UQ). This is accomplished by computing the relative root squared error in: (i) parameter space, by comparing the estimates from all 3 methods to the ground-truth parameter values that generated the data (Eq. (9)), and (ii) output space, by comparing the data predictions obtained with the different parameter estimates to the competition data (Eq. (10)).

$$\text{RRSE}_{\text{par}} = \sqrt{\sum_{i=1}^{d} \left(\frac{\theta_i - \hat{\theta}_i}{\theta_i} \right)^2}, \tag{9}$$

where $\boldsymbol{\theta}$ is the ground-truth parameter vector of dimension $d \times 1$ and $\hat{\boldsymbol{\theta}}$ is the estimated parameter vector.

$$\text{RRSE}_{\text{output}} = \sqrt{\sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{\max(\mathbf{y})} \right)^2}, \tag{10}$$

where $\mathbf{y}$ is the competition signal of dimension $n \times 1$ and $\hat{\mathbf{y}}$ is the estimated signal. $\text{RRSE}_{\text{output}}$ was computed for each of the three individual signals (one pressure and two flow signals).

6.2. Pulmonary model - accuracy and UQ

For the pulmonary model, the competition data were noisy, and so the assessment was conducted with respect to both accuracy and UQ in parameter and output space. The organisers obtained "ground-truth" MCMC samples by running a simulator-based AM sampler [51] using custom MATLAB scripts. Further details are presented in Section 3 of the Supplement.

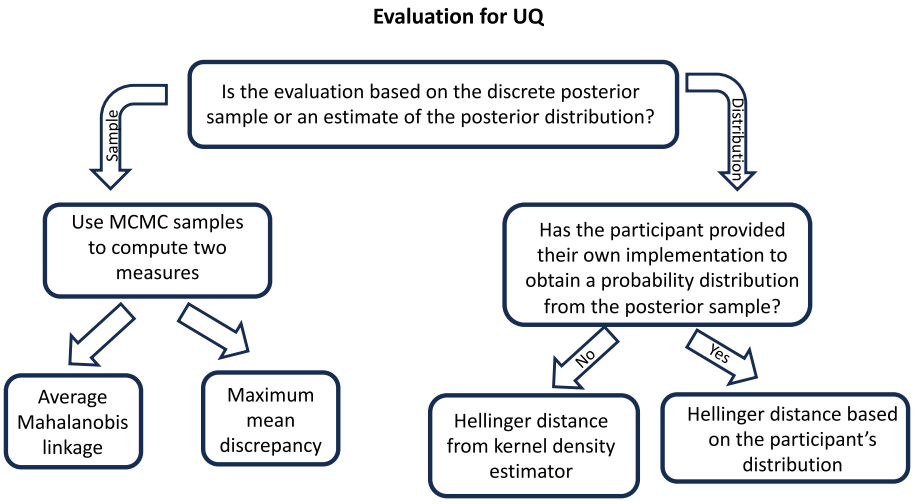


Fig. 4. Overview of the assessment criteria. See text for details.

6.2.1. Parameter space

Let $p(\boldsymbol{\theta}|\mathbf{y})$ denote the true posterior distribution of the parameter vector $\boldsymbol{\theta}$ given data $\mathbf{y}$, approximated using the organisers' long run simulator-based MCMC sampler, and $q(\boldsymbol{\theta}|\mathbf{y})$ the estimated posterior distribution obtained with the methods in Table 3.

Accuracy: Participants were asked to submit a parameter estimate based on $q(\boldsymbol{\theta}|\mathbf{y})$ (either the mean posterior or maximum aposteriori point), which we denote by $\hat{\boldsymbol{\theta}}$. We define the accuracy metric as the conditional probability of the estimate given the set of simulator-based MCMC samples, X, denoted by $p(\hat{\boldsymbol{\theta}}|X)$. The best submission with respect to accuracy is the one with the lowest negative log probability value, i.e. $-\log(p(\hat{\boldsymbol{\theta}}|X))$.

To compute $p(\hat{\boldsymbol{\theta}}|X)$, the organisers needed to estimate the density from the MCMC samples, for which a standard approach based on a kernel density estimator (KDE) [52] was adopted, with two ways of estimating the bandwidth, i.e. Silverman's rule [53] and leave-one-out cross validation (LOO CV) [54]. This is a classical textbook criterion; however, more advanced methods have been proposed in the more recent literature, e.g. based on GPs [55,56]. The organisers thus decided to invite the participants to submit their own code for their preferred density estimation method.

UQ: To assess UQ, the organisers have used three criteria: (i) a divergence-based measure (Hellinger distance [57]) for which posterior densities need estimating, and (ii) two criteria based on the submitted posterior samples: average linkage, based on a Mahalanobis distance, and maximum mean discrepancy (MMD) [58], based on a linear (Mahalanobis) kernel. An overview of the different assessment criteria is provided in Fig. 4, and we provide the methodological details below.

- Hellinger distance: The Hellinger distance (HD) [57] is a measure based on divergence between two distributions, i.e. between $p(\boldsymbol{\theta}|\mathbf{y})$ and $q(\boldsymbol{\theta}|\mathbf{y})$, that are estimated using methods described above. HD is defined as the L2 norm between $\sqrt{p(\cdot)}$ and $\sqrt{q(\cdot)}$:

$$\mathrm{HD}^2[p,q] = \frac{1}{2}\int\left(\sqrt{p(\boldsymbol{\theta}|\mathbf{y})}-\sqrt{q(\boldsymbol{\theta}|\mathbf{y})}\right)^2 d\boldsymbol{\theta} = \frac{1}{2}\int\left(1-\sqrt{\frac{q(\boldsymbol{\theta}|\mathbf{y})}{p(\boldsymbol{\theta}|\mathbf{y})}}\right)^2 p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \tag{11}$$

  The best submission is the one with the lowest Hellinger distance. The Hellinger distance was used instead of the more widely known Kullback–Leibler divergence [59] because in practice the organisers found that the effective support of the true posterior distribution may not include the samples generated with some of the methods, leading to a singularity. HD was thus used as a numerically more robust alternative.
- Mahalanobis distance: Consider two sets of posterior samples $X = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ and $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$, where X is the set of posterior samples from the true mathematical model (the simulator), and Z is the set of posterior samples from the surrogate model (the emulator). A natural discrepancy measure is the normalised sum over all pairwise Mahalanobis distances between elements in set X and those in set Z:

$$\mathrm{MD} = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(\mathbf{x}_m-\mathbf{z}_n)^\top\mathbf{H}(\mathbf{x}_m-\mathbf{z}_n) = \frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_m^\top\mathbf{H}\mathbf{x}_m + \frac{1}{N}\sum_{n=1}^{N}\mathbf{z}_n^\top\mathbf{H}\mathbf{z}_n - \frac{2}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathbf{x}_m^\top\mathbf{H}\mathbf{z}_n, \tag{12}$$

  where $\mathbf{H}$ is the inverse covariance matrix of the reference distribution. In our case, the reference distribution consists of the posterior samples from the true mathematical model (the simulator). MD is a linkage method to quantify the difference between two samples, and a submission with the lowest MD score is preferred.
- MMD: Another discrepancy measure used is the MMD [58], defined as follows:

$$\mathrm{MMD} = \frac{1}{M(M-1)}\sum_{m=1}^{M}\sum_{m'\neq m}^{M}\mathbf{x}_{m'}^\top\mathbf{H}\mathbf{x}_m + \frac{1}{N(N-1)}\sum_{n=1}^{N}\sum_{n'\neq n}^{N}\mathbf{z}_{n'}^\top\mathbf{H}\mathbf{z}_n - \frac{2}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathbf{x}_m^\top\mathbf{H}\mathbf{z}_n. \tag{13}$$

Table 4
Systemic model: root relative square error in parameter space (Eq. (9)). The lowest value (in bold) is best.

Method I	Method II	Method III
0.09	$\mathbf{9.4 \cdot 10^{-7}}$	0.02

Table 5
Systemic model: root relative square error in output space (Eq. (10)) for every signal (S) displayed in Fig. 3(a). The lowest value (in bold) is best.

	Method I	Method II	Method III
S1	0.005	$\mathbf{9.6 \cdot 10^{-8}}$	0.001
S2	0.013	$\mathbf{1.8 \cdot 10^{-7}}$	0.003
S3	0.069	$\mathbf{1.1 \cdot 10^{-6}}$	0.008
S4	0.017	$\mathbf{2.7 \cdot 10^{-7}}$	0.002
S5	0.062	$\mathbf{1.4 \cdot 10^{-6}}$	0.013
S6	0.014	$\mathbf{3.9 \cdot 10^{-7}}$	0.002
S7	0.022	$\mathbf{5.1 \cdot 10^{-7}}$	0.003
S8	0.002	$\mathbf{3.8 \cdot 10^{-8}}$	0.0002

Although originally MMD was developed as a test statistic for rejecting the null hypothesis that two samples are from the same distribution, in this work it is used as a method discriminating between the different competition submissions, i.e. the submission with the lowest MMD score is preferred.
MMD is a more pessimistic measure than MD as it accounts for cross-correlations between terms in the same set, and hence more terms are being added (note the double sum in the first two terms of the expression in Eq. (13)).

6.2.2. Output space

Uncertainty in parameter space propagates to output space. Hence, for every posterior parameter sample, we can run the mathematical model and obtain the corresponding (multivariate – 512D –) signal. Repeating this for the entire set of posterior parameter samples leads to an ensemble of signals.

The procedure to evaluate accuracy and UQ in output space follows closely that in parameter space. While the parameter space is 4D, the output space is 512D, which means that while estimating densities is tractable for a 4D space, this is not the case for 512D. For this reason, the organisers applied some of the assessment evaluation in a univariate sense, i.e. for the maximum (systolic) and minimum (diastolic) points of the time series, separately for every output signal. This applied to the measures that used density estimation, namely the accuracy-based measure of conditional probability and UQ-based measure of HD. In contrast, MD and MMD has been computed both univariately (for the maximum and minimum points of the time series) and multivariately (512D).

7. Assessment results

Below we present evaluation results for all three competition methods on both CFD models.

7.1. Systemic model

7.1.1. Parameter space

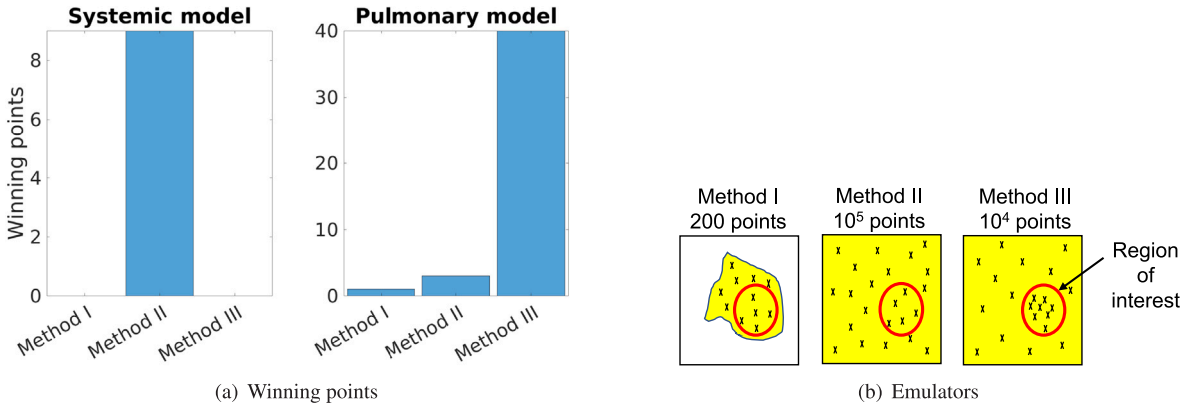
Table 4 shows the value of the root relative square error in parameter space (Eq. (9)). We notice that method II records the lowest value, which is substantially lower than the values of the other two methods. Moreover, method III has the second best performance, followed by method I.

7.1.2. Output space

Next, we present the root relative square error in output space for each of the 8 signals, computed using Eq. (10), displayed in Table 5. Method II consistently records the lowest value for every signal. Additionally, method III is second best, followed by method I.

7.1.3. Systemic: summary of winning points

In conclusion, by collating the results from each individual assessment, we note that method II has won all the 9 points (see left-hand graph in Fig. 5(a)), and is therefore the winning method for the systemic model.



(a) Winning points (b) Emulators

Fig. 5. Panel (a): Summary of competition scored assessment points. Panel (b): Emulators: for method I, the emulator at the highest wave number is zoomed in on the most promising region (200 training points), with no coverage outside this region, method II's emulator covers the entire parameter space uniformly using a very large number of training points (10^5), and method III's emulator covers the entire parameter space (10^4 training points), with a denser coverage on the most promising region.

Table 6
Pulmonary model: negative log probability of each method's parameter estimate $\hat{\theta}$ given the simulator parameter MCMC samples X, i.e. $-\log(p(\hat{\theta}|X))$ for the cases where the bandwidth for kernel density estimation based on the MCMC samples is selected using Silverman's rule (first entry) and leave-one-out cross-validation (second entry). The lowest value (in bold) is best.

	Method I	Method II	Method III
Silverman/CV	175/536	17/31	**6/6**

Table 7
Pulmonary model: agreement in uncertainty quantification in parameter space between the competition methods and the "ground-truth" assessed using multivariate (4-dimensional) HD (Helinger distance) – Eq. (11), MD (Mahalanobis distance) – Eq. (12), MMD (maximum mean discrepancy) – Eq. (13). HD is calculated for the cases where the bandwidth for kernel density estimation based on the posterior samples is selected using Silverman's rule (first entry) and leave-one-out cross-validation (second entry). The lowest value (in bold) is best.

	Method I	Method II	Method III
HD	0.79/0.86	0.63/0.70	**0.30/0.34**
MD	7.49	3.23	**2.09**
MMD	1.78	0.62	**0.06**

7.2. Pulmonary model

7.2.1. Parameter space

Accuracy: Table 6 shows the negative log probability of each method's parameter estimate $\hat{\theta}$ given the simulator parameter MCMC samples X, i.e. $-\log(p(\hat{\theta}|X))$ for the cases where the bandwidth for KDE based on the MCMC samples is selected using Silverman's rule and CV. We notice that method III records the lowest negative log probability, followed by method II, and method I comes last. These findings align for both cases when the bandwidth was found with Silverman's rule and CV.

UQ: As previously mentioned, 3 criteria have been used to assess agreement in UQ between the competition methods and the ground-truth, obtained by running MCMC with the simulator: HD (Eq. (11), computed with KDE with the bandwidth selected with Silverman's rule and CV), MD (Eq. (12)) and MMD (Eq. (13)), calculated in multivariate (4D) parameter space. Table 7 shows the HD, MD and MMD scores for each of the 3 methods. We observe that method III performs systematically best according to all 3 criteria, as it records the lowest scores, followed by method II, and lastly, method I.

We also visualise the univariate parameter densities obtained with KDE based on posterior samples for all methods, and the ground-truth densities are also superimposed in Fig. 6. While in a multivariate sense, it is clear that method III has the best performance, in a univariate sense, the performance depends on the parameter. For example, for parameters f_3^{MV} and α, the agreement in densities between the simulator and method II appears to be largest, while for parameters lrr_{A} and lrr_{V}, the largest overlap of the simulator densities is with densities from method III. The densities from method I look overly wide, thus the uncertainty is overestimated.

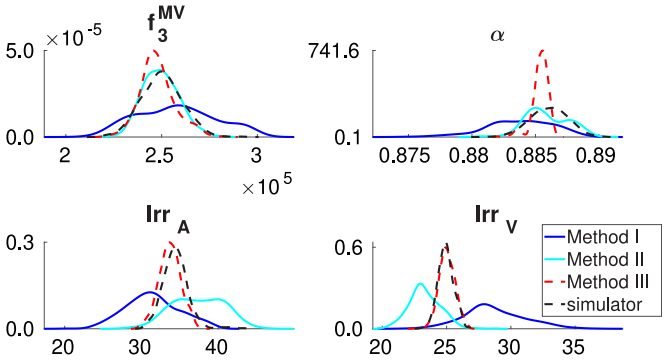


Fig. 6. Pulmonary model: univariate densities obtained with kernel density estimation based on posterior samples for all 4 biophysical parameters. Densities from all methods are shown and the ground-truth density obtained with the simulator is superimposed.

Table 8
Systemic model: negative log probability of each method's systolic and diastolic estimates for every signal (S) displayed in Fig. 3(b) given the simulator MCMC samples for the cases where the bandwidth for kernel density estimation is selected using Silverman's rule (first entry) and leave-one-out cross-validation (second entry). The lowest value (in bold) is best.

		Method I	Method II	Method III
Systolic	S1	7.05/6.91	**0.65/0.65**	0.75/0.75
	S2	0.71/0.71	0.85/0.84	**0.43/0.47**
	S3	**1.60/1.62**	2.12/2.12	1.63/1.66
Diastolic	S1	−0.54/ − 0.55	−0.21/ − 0.22	−0.79/ − 0.81
	S2	0.32/0.31	0.17/0.18	−0.22/ − 0.19
	S3	1.78/1.74	1.03/1.06	**0.79/0.81**

7.2.2. Output space

Accuracy: Table 8 shows the negative log probability of the methods' systolic and diastolic estimate for every signal given the simulator posterior samples for the cases where the KDE bandwidth is selected using both Silverman's rule and CV. We observe that while for the diastolic point, method III systematically records the lowest negative log probability values across all signals, for the systolic point there is no clear winner, the performance depending on the signal, and these findings agree for both Silverman and CV.

UQ: To assess agreement in UQ between the densities generated with the competition methods and the ground-truth densities, obtained from a simulator-based MCMC simulation, the organisers have calculated HD (Eq. (11)) for the systolic and diastolic points, MD (Eq. (12)) and MMD (Eq. (13)) for the systolic and diastolic points (and all the other time points univariately), as well as for the whole time series of every signal (in a multivariate 512D output space).

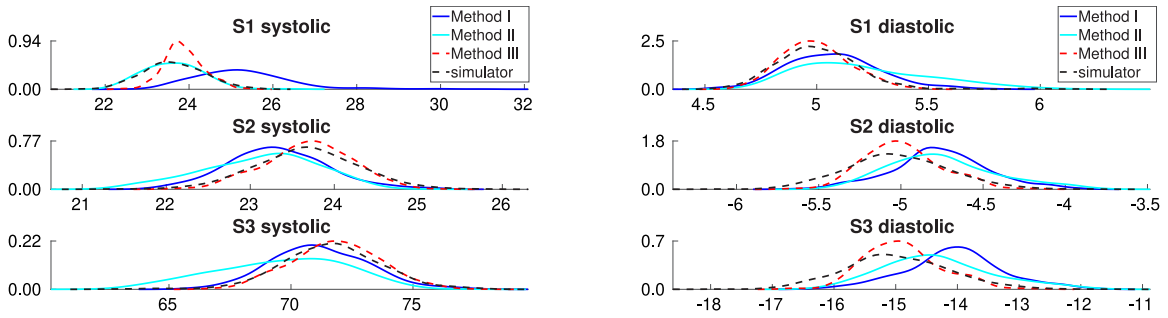
Table 9 shows the HD, MD and MMD scores for the systolic and diastolic points of every signal for each of the 3 methods. Method III tends to have the best performance according to all 3 criteria. The HD and MMD scores are in perfect agreement with each other in terms of the method ranking. The MD score is in a slight disagreement for the systolic point of signal 1, S1, while agreeing with HD and MMD for everything else. A possible explanation for this disagreement is that MMD is a more pessimistic measure, as explained in more details under Eq. (13).

Furthermore, Fig. 7 shows the univariate densities of the systolic (panel (a)) and diastolic (panel (b)) points obtained with KDE based on posterior samples for all methods, and the ground-truth densities are superimposed. The findings from the visual inspection are on a par with the objective measures computed above. More specifically, it appears that method III's densities are closest to the simulator's densities for all inspected quantities, except the systolic point of signal 1 (S1), where method II's density is in nearly perfect agreement with the simulator's density.

Additionally, Fig. 8 illustrates the MD and MMD scores calculated univariately at each time point in the time series, and Table 10 shows the average univariate scores (average taken over all time points), as well as the multivariate scores. We observe that method III systematically records the lowest average univariate and multivariate scores (Table 10), and tends to have the lowest scores for most time points (Fig. 8). When comparing the performance of methods I and II, there is no clear better performance as that is dependent on the signal and the time portion of the signal. For example for signal 1, the lower scores are recorded for method II in the middle of the series (between time points 100 and 300), but method I has lower scores everywhere else (see Fig. 8(a) and (d)). The average univariate and multivariate MD and MMD scores also agree in ranking between methods I and II, except for signal 3 (Table 10), for which the multivariate score prefers method II over I, but the opposite holds for the univariate case. This disagreement may be due to the extra correlations that are captured by the multivariate score compared to the univariate score.

Table 9
Pulmonary model: agreement in uncertainty quantification in output space for the systolic and diastolic points of all signals (S) displayed in Fig. 3(b), between the results from the three competition methods and the "ground-truth" assessed using HD (Hellinger distance) – Eq. (11), MD (Mahalanobis distance) – Eq. (12), MMD (maximum mean discrepancy) – Eq. (13). HD is calculated for the cases where the bandwidth for kernel density estimation based on the posterior samples is selected using Silverman's rule (first entry) and leave-one-out cross-validation (second entry). The lowest value (in bold) is best.

			Method I	Method II	Method III
HD	Systolic	S1	0.33/0.33	**0.005/0.005**	0.07/0.07
		S2	0.04/0.04	0.08/0.08	**0.009/0.01**
		S3	0.013/0.013	0.09/0.09	**0.010/0.010**
	Diastolic	S1	0.03/0.03	0.09/0.09	**0.01/0.01**
		S2	0.09/0.09	0.09/0.09	**0.02/0.03**
		S3	0.14/0.14	0.07/0.07	**0.03/0.04**
MD	Systolic	S1	8.93	1.98	**1.54**
		S2	2.21	2.97	**1.69**
		S3	1.95	3.54	**1.75**
	Diastolic	S1	2.19	4.19	**1.58**
		S2	2.55	2.80	**1.57**
		S3	3.04	2.65	**1.55**
MMD	Systolic	S1	5.18	−0.003	0.11
		S2	0.29	0.75	−0.001
		S3	0.04	0.93	**0.006**
	Diastolic	S1	0.09	1.01	**0.002**
		S2	0.67	0.72	**0.007**
		S3	1.17	0.58	**0.008**



(a) Systolic point (b) Diastolic point

Fig. 7. Pulmonary model: univariate densities obtained with kernel density estimation based on posterior samples for the systolic (panel (a)) and diastolic (panel (b)) points of every signal (S) displayed in Fig. 3(b). Densities from all methods are shown and the ground-truth density obtained with the simulator is superimposed.

7.2.3. Pulmonary: summary of winning points

In conclusion, by combining the results from each individual assessment, method III has won the majority of the points, as seen in the right-hand graph in Fig. 5(a), and is therefore the winning method for the pulmonary challenge.

7.3. Intuitive method comparison metrics

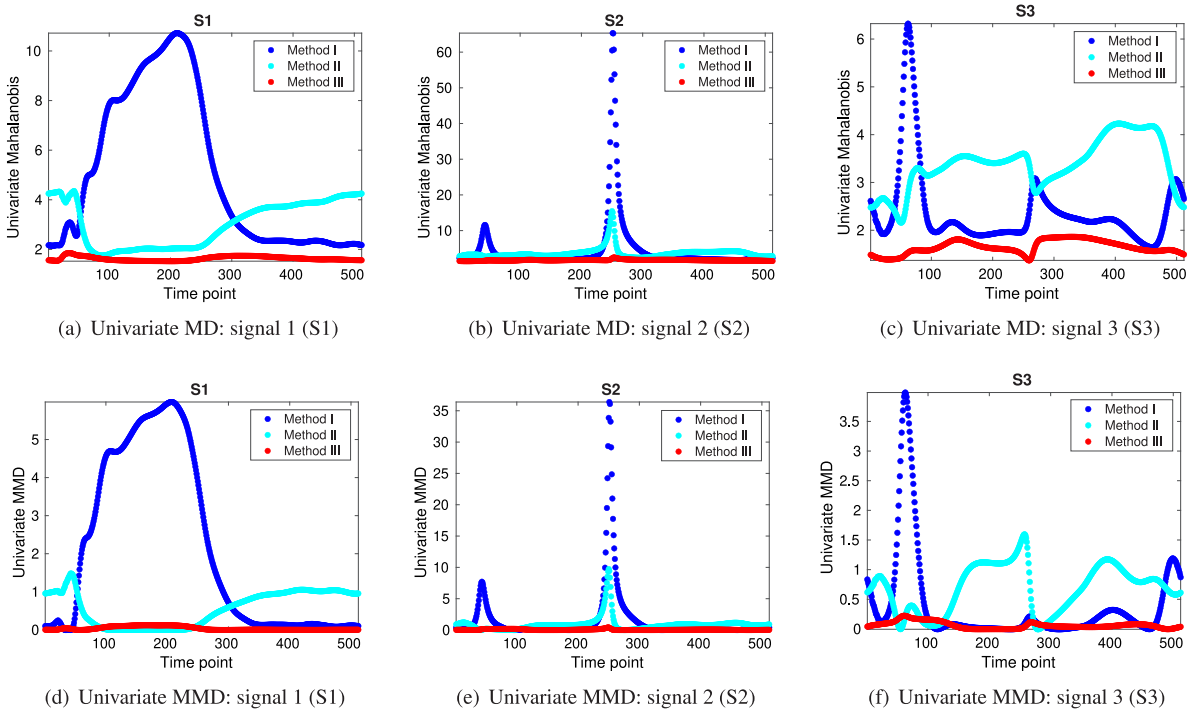
Our evaluation metric based on the Hellinger distance quantifies how well the predicted posterior distribution agrees with the gold standard distribution, and it is used for model performance ranking. However, Hellinger distances do not give us a good intuition for model performance in absolute or relative terms. For that reason we also include more intuitive evaluation metrics based on point estimates (rather than entire distributions), such as the percent deviation from the "ground-truth" in parameter space, defined as

$$\mathrm{PD} = \frac{1}{d}\sum_{i=1}^{d}\left|\frac{p_i - \hat{p}_i}{p_i}\right|, \tag{14}$$

where p_i defines the "ground-truth", and $\hat{p}_i$ defines the estimate. The percent deviation is calculated for the parameter estimates in the systemic model, and for the parameter posterior mean and variance in the pulmonary model. While the percent deviation has the advantage of being intuitive, it does not provide an assessment of the complete distribution for the pulmonary challenge (i.e. only

Table 10
Pulmonary model: average univariate Mahalanobis distance (MD) – Eq. (12) and maximum mean discrepancy (MMD) – Eq. (13) scores (average taken over all time points in the time series) and multivariate (512D) scores over the whole time series for every signal (S) displayed in Fig. 3(b) and method. The lowest value (in bold) is best.

			Method I	Method II	Method III
MD	Average univariate	S1	5.03	3.03	**1.63**
		S2	4.41	3.68	**1.71**
		S3	2.38	3.38	**1.64**
	Multivariate	S1	46 858	731	**49.9**
		S2	34 850	8830	**77.5**
		S3	26 461	3414	**61.5**
MMD	Average univariate	S1	2.1	0.5	**0.03**
		S2	1.54	0.86	**0.05**
		S3	0.37	0.68	**0.06**
	Multivariate	S1	7756	245	**2.69**
		S2	15 477	4166	**6.47**
		S3	12 906	1560	**5.51**



(a) Univariate MD: signal 1 (S1) (b) Univariate MD: signal 2 (S2) (c) Univariate MD: signal 3 (S3)

(d) Univariate MMD: signal 1 (S1) (e) Univariate MMD: signal 2 (S2) (f) Univariate MMD: signal 3 (S3)

Fig. 8. Pulmonary model: Mahalanobis distance (MD) – Eq. (12) and maximum mean discrepancy (MMD) – Eq. (13) scores computed univariately for each time point in the time series of every signal (S) displayed in Fig. 3(b) for all methods. Lower values are better.

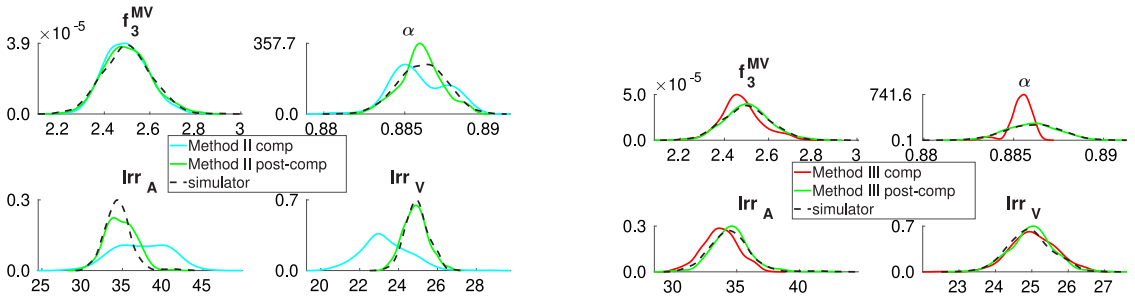
for the first two moments of the distribution, mean and variance), and hence it is not included as a formal competition assessment criterion. In Table 11, we notice that for the systemic challenge, the relative deviation from the "ground-truth" in parameter space ranges from 10^{-5}% (highest-performing method II) to 3% (lowest-performing method I). For the pulmonary challenge, the performance ranges from 0.9% (highest-performing method III) to 7% (lowest-performing method I) deviation for the parameter posterior mean, and from 35% (method III) to 570% (method I) deviation for the variance.

7.4. Post-competition assessment

After the competition, the contestants were given the opportunity to improve their methods upon inspection of their performance scores. For the pulmonary challenge, method II was improved by fitting a larger NN, and using a modified loss function during NN fitting to include the correct noise model released by the organisers in order to form an idealised scenario (see Section 5.2.3 for details). Additionally, for both challenges, method III was improved by fitting a second emulator with an improved GP hyperparameter search based on a gradient-based optimiser while using the entire training dataset, and by using an inference scheme

Table 11
Percent deviation from "ground-truth" of parameter estimates (systemic model), and of parameter posterior mean and variance (pulmonary model), calculated using Eq. (14).

		Method I	Method II	Method III
Systemic	Estimates	3%	10^{-5}%	0.6%
Pulmonary	Posterior mean	7%	4%	0.9%
	Posterior variance	570%	160%	35%



(a) Method II (b) Method III

Fig. 9. Pulmonary model (post-competition): univariate posterior densities obtained with kernel density estimation based on posterior samples for all 4 biophysical parameters for methods II and III. Post-competition densities were obtained after improvements were made to the methods, see Sections 5.2 and 5.3 for details.

based on gradient-based optimisation for the systemic model and a tempered likelihood in the MCMC exploratory phase for the pulmonary model (see Section 5.3.3 for details). Fig. 9 shows the post-competition univariate densities for the pulmonary challenge for both methods II and III. A clear improvement over the competition-submitted densities is observed for both methods, with the post-competition densities following the simulator densities much more closely. For the systemic challenge, post-competition method III also leads to more accurate parameter estimates than the competition entry, with a root relative square error in parameter space of $9.3 \cdot 10^{-4}$.

7.5. Emulator accuracy

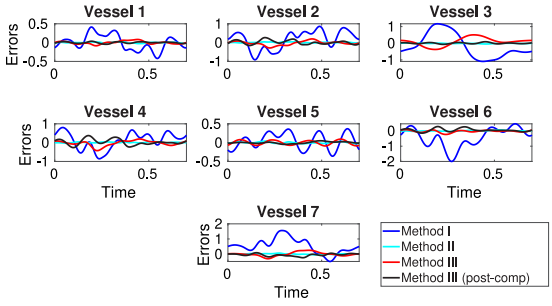
To understand how the emulator accuracy compares between methods[1], post-competition, the organisers provided test simulator outputs generated with parameter samples in the high-posterior area. For the pulmonary model, these were draws from the simulator-based ground-truth distribution. For the systemic model, for which there was no uncertainty in the estimated parameter values due to noise-free data, the competition data only were provided as a test output, to avoid creating a disadvantage for the History Matching based emulator, which focuses on a subset of the parameter space containing likely data-generating parameter values. Providing test outputs generated with parameter values drawn from a space-filling design across the entire parameter space would be inappropriate for the History Matching emulator. Fig. 10(a) shows that for the systemic model, method I records the largest errors (sum of squared errors, RSS = 0.7), followed by method III, (RSS = 0.06 and RSS post-competition = 0.006), and method II has the lowest errors (RSS = 0.002). Hence, method II, which is the winning method in the inference portion of the systemic model, has also produced the most accurate emulator. A similar finding is obtained for the pulmonary challenge, for which Fig. 10(b) shows that method II records the lowest errors, followed by method III, and lastly, method I. We also notice that post-competition, the improved methods II and III have generated more accurate predictions compared to the competition predictions.

8. Discussion

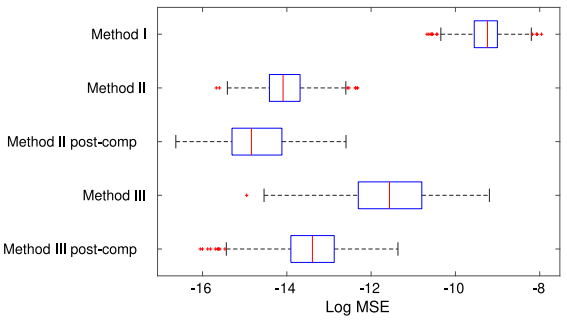
8.1. Method differences

We proceed to discuss conceptual differences between the three methods employed for emulation, estimation and UQ. All 3 emulation methods are based on a space-filling design constructed prior to the data being released. Post-data release, emulators thus constructed are used differently depending on the inference method of choice. Method I uses History Matching as an inference scheme, which given the data, refines the original emulator sequentially, in waves, i.e. emulator at wave $i+1$ is zoomed in on a subset of the parameter space containing parameter values that are more likely to have generated the data; in this region the $(i+1)$th wave emulator is more accurate than ith wave emulator with respect to an implausibility criteria, see Section 5.1.2 for details. The

[1] For details about the number of training points for individual emulators, see Sections 5 and 8.1.



(a) Systemic model: emulator accuracy on test data.



(b) Pulmonary model: emulator accuracy on test data.

Fig. 10. Panel (a): Systemic model: vessel-specific errors, calculated as the difference between simulator output and emulator predictions for the competition parameter values shown in Table 1 for all 3 methods (including a post-competition submission corresponding to an improved method III). Panel (b): Pulmonary model: distribution of log mean square errors (MSE) between simulator output and emulator predictions for 500 simulator-based posterior parameter samples for all 3 methods (including post-competition submissions from improved methods II and III); MSE is computed as the sum of squared errors divided by the number of points in the output time series, and summed for all vessels.

training points of the emulator at wave $i+1$ may include a subset of the training points of emulator i (those that have satisfied an implausibility criteria), along with newly sampled training points that are non-implausible. The number of training points at every wave is 100 or 200 (depending on the CFD model). Method II uses the original emulator built prior to seeing the data, i.e. the emulator is not refined to be data-specific, and the inference scheme used does not change the emulator. The number of training points is 10^5. Method III uses an emulation approach that builds on the original emulator after seeing the data, i.e. new training points are added to the list of points for the original emulator. These are points that are likely to have generated the data, as given by the inference scheme. For example, for the systemic model, optimised parameter values obtained from multiple optimisation rounds were added as training points, see Section 5.3.2 for details. All the original training points are a subset of the final set of training points, and the number of training points is 10^4 or $1.5 \cdot 10^4$ (depending on the CFD model).

To summarise, the difference between methods is that while for method I, the emulator at the highest wave number is zoomed in on the most promising region, with no coverage outside this region, method II's emulator covers the entire parameter space uniformly using a very large number of training points, and method III's emulator covers the entire parameter space, with a denser coverage of the most promising region. These differences in emulators are illustrated in Fig. 5(b).

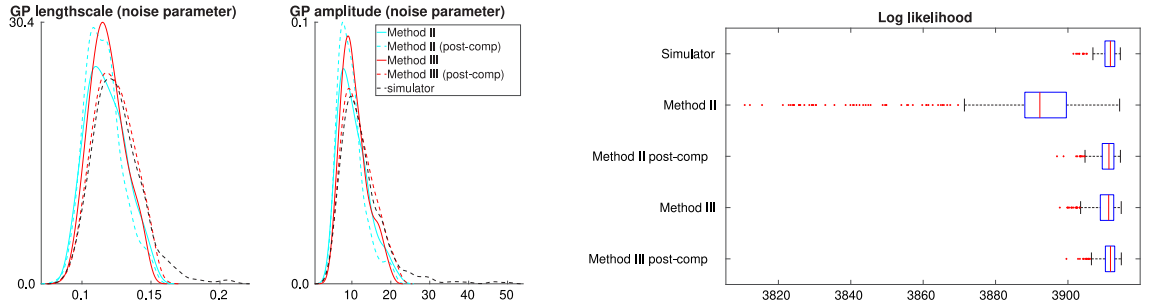
There are two main reasons for the different performance between methods: (i) emulator accuracy and (ii) inference and UQ scheme.

Emulator effect: In principle, the very large number of training points used by method II suggests that the accuracy of the corresponding emulator is highest. While this may be true when looking at the whole parameter space, it does not necessarily hold for the region of interest that contains parameters likely to have generated the data, since methods used different inference schemes, which may help refine the emulator, as explained above. Investigations into emulator accuracy have revealed that for both systemic and pulmonary challenges, the deep residual NN used by method II is indeed most accurate in the region of interest, followed by method III's GP emulator, and lastly method I's History Matching-based GP emulator. In addition, post-competition investigations have revealed that the improved methods II and III have a higher predictive accuracy, i.e. the improved method II features a larger, more complex NN emulator (details in Section 5.2.1), while the improved method III uses a different, gradient-based optimisation scheme for the GP kernel hyperparameters (details in Section 5.3.1).

Inference and UQ scheme: All 3 methods use established inference techniques, widely adopted by the machine learning community. For the systemic model, to infer the parameter values, method II uses simulated annealing (on the emulator), and method III uses greedy optimisation with multiple restarts (on the emulator). In contrast, method I uses an approach based on "best input" out of a fixed number of samples, which is History Matching-based *discrete* optimisation (see Section 5.1.2). It appears that this is sub-optimal compared to the *continuous* optimisation schemes of the other two methods. In principle, the History Matching emulator could be improved by having further wave refinements, to allow continuously zooming in on the parameter region of interest. However, the inference task had to be conducted within an allocated three-hours limit (which was unknown to the contestants prior to releasing the exact, model-specific competition challenges and the data), which limited the number of feasible wave refinements and appears to have rendered History Matching suboptimal for this task.

In addition, although in method I only the flow data was used for emulation and inference (i.e. the two pressure points were excluded), post-competition exploration revealed that under the same competition method settings used in the competition, the inclusion of pressure did not improve the accuracy of the results, a finding which is in line with that from method II (see Table 6 in the Supplement). This result is intuitive given that the flow data are much more abundant, hence in principle more informative than pressure data.

For the pulmonary model, developers of methods II and III have used MCMC (on the emulator) to generate samples from the posterior distribution in a Bayesian framework, while the developer of method I has obtained samples from History Matching in combination with a rejection sampling mechanism (details in Section 5.1.2). The constructed History Matching emulator appears to



(a) Inferred noise model parameters (b) Corresponding log likelihood

Fig. 11. Panel (a): univariate posterior densities obtained with kernel density estimation based on posterior samples for the 2 noise model parameters, drawn using the simulator or emulators from methods II and III (competition and post-competition). Panel (b): The corresponding log likelihood obtained with the posterior samples of the biophysical and noise parameter values, see Eq. (15).

be inflicted by a large prediction uncertainty, which may help explain the generally overdispersed densities obtained. As future work, a few more wave refinements may be performed, and after building the last wave emulator, an MCMC sampler may be run [60,61]. Additionally, unlike methods II and III, that utilise MCMC to allow for uncertainty in the error parameters (besides the biophysical parameters), method I uses fixed error parameters, which may potentially bias the inferred biophysical parameters. In part, this may be due to the organisers releasing the noise model at the same time as the data, and so, if the noise model had been known in advance, a different, more suitable approach may have been taken.

Emulator versus inference discrepancy: A peculiar finding is that the method with the apparently highest emulation accuracy is not guaranteed to be the method with the best inference performance. For the systemic model, for which inference was based on noise-free data, emulator accuracy is positively correlated with inference accuracy, i.e. the method with the most accurate emulator also records the highest inference accuracy. For the pulmonary model, for which inference was performed from noisy data, method II showed the highest emulation accuracy, according to the criteria discussed in the previous subsection on "emulator effect", but that was not reflected in the inference accuracy, for which method III was best. A potential explanation for this discrepancy is the influence of the correlated noise model parameters (i.e. GP lengthscale and amplitude, see Section 3.2 for details). The noise model parameters are nuisance parameters, whose inference was not directly assessed in the competition. Fig. 11(a) shows univariate posterior densities (kernel density estimation plots) of the nuisance parameters constructed from MCMC samples drawn with the simulator and with emulators from methods II and III (competition and post-competition). The plot reveals that there are clear differences in the inferred nuisance parameters between methods II and III, which has repercussions on the inference accuracy of the biophysical parameters. In Fig. 11(b) we also present the log likelihood obtained with MCMC samples of the biophysical and noise model parameters,

$$\mathcal{L}(\mathbf{y}|\boldsymbol{\theta}, \mathbf{C}(\boldsymbol{\xi})) = \log(\det(2\pi\mathbf{C})^{-\frac{1}{2}}) - 0.5(\mathbf{y} - f(\boldsymbol{\theta}))^{\mathsf{T}}\mathbf{C}^{-1}(\boldsymbol{\xi})(\mathbf{y} - f(\boldsymbol{\theta})), \tag{15}$$

where $\mathbf{y}$ is the competition data, $\boldsymbol{\theta}$ are emulator or simulator-drawn biophysical parameter samples, which are passed through the simulator to produce the corresponding simulator output, $f(\boldsymbol{\theta})$, and $\mathbf{C}(\boldsymbol{\xi})$ is the noise covariance matrix obtained using the emulator or simulator-inferred noise parameters, $\boldsymbol{\xi}$, see Section 3.2 for details on how to obtain $\mathbf{C}$ from $\boldsymbol{\xi}$.

From Fig. 11(b), it is apparent that method III has a larger (better) log likelihood than method II, that the log likelihood obtained with method III is more similar to that of the simulator, i.e. the ground truth, and that post-competition log likelihoods are larger than competition log likelihoods for both methods II and III.

8.2. Exact inference

Although some methods returned emulator-based posteriors that resembled the reference posterior quite closely (see univariate densities in Fig. 6), some discrepancies were present. For instance, Fig. 6 shows that method III's posterior for the parameter α was over-confident, and hence using these emulator samples may lead to ruling out a true parameter setting, while the other methods tended to be less confident than the reference posterior. Given the high-dimensional output and the GP observation model used, the likelihood function has been found to be very sensitive to small discrepancies between the emulator and simulator, implying that the emulator-based MCMC sampler has returned densities having some form of discrepancy to the simulator densities. In Fig. 9 we have shown that it is possible to build emulators based on which MCMC samplers can produce univariate densities resembling the reference densities much more closely, although some minor discrepancies remain. This clearly illustrates the difficulty in obtaining emulator densities that are in perfect agreement with the simulator densities.

This problem can be overcome by running an emulation-based MCMC with a correction step based on the simulator, for example using a delayed acceptance scheme [61,62], which is a two-stage acceptance procedure, with two separate acceptance/rejection decisions. The first decision is a computationally fast pre-filter step based on the surrogate model, which upon rejection of a proposed

new parameter avoids carrying out the computationally expensive second step based on the simulator. Such an approach ensures that samples are drawn from a distribution that asymptotically converges to the true posterior distribution, and can significantly lower the number of simulator evaluations when compared to a simulator-based only MCMC (i.e. only a couple thousand simulator evaluations may be needed [61]), making the procedure viable to be run within the allocated one week.

8.3. Model discrepancy

The organisers have designed the simulated data such that they are representative of real clinical data. The correlated noise in the pulmonary model reflects typical noise found in real data and captures *in principle* any model discrepancy between the simulator and the real system for the cases when real data are used [63]. This is based on the fact that any discrepancy between the true signal and the model prediction can be modelled with a Gaussian process due to its universal approximation capability [64]. Model discrepancy may be due to numerical errors (e.g., numerical integration of the PDEs) or simplifying model assumptions (e.g., purely elastic vessel walls, or the 1D model simplification) [12]. However, in a real application, the adequate kernel for modelling the model discrepancy would not be known and would have to be learned by a combination of inference from data and systematic integration of physical constraints into the kernel design [65]. For the sake of keeping the assessment of method performance as part of the competition sufficiently simple, we have chosen a kernel of a standard form and assumed it to be known. Consequently, the posterior uncertainties presented in this paper may be lower than what would be expected in a real clinical application.

8.4. Clinical translation & future work

A long-term goal of this work is to develop a real-time clinical decision support system relying on the combination of physical, mathematical and statistical modelling of (patho)physiological systems, to enable personalised healthcare. To this end, the competition has provided a platform to explore state-of-the-art emulation and inference techniques from Statistics and Machine Learning, and has aimed to identify the best-performing methods. While this study is a stepping stone towards clinical translation, it naturally comes with shortcomings, which may be addressed in future studies. For real-data studies, some limitations include: (i) Model simplifications, e.g. 1D simplification instead of more physiologically realistic, 3D CFD models, which are associated with much higher computing costs. To circumvent this issue, a multifidelity emulation approach based on the mapping of 1D and 3D models may be adopted [66]; (ii) Parameter fixing based on sensitivity analysis: although fixing some non-identifiable model parameters to empirical or nominal values reduces the parameter dimensionality, and simplifies the emulation and inference task, it may bias the inference parameters. Hence, future work could address the identifiability issues by the integration of physiological prior knowledge [11]; (iii) Uncertainty in vessel network geometry, which may be captured by specifically including the geometry parameters in the emulation and inference scheme. This will increase the parameter complexity, potentially leading to a much larger number of training points required to train the emulator to ensure a dense enough coverage of the parameter space. To deal with the numerical problems that arise when using emulators for large data sets, other methods could be employed, such as local GPs [67], sparse GPs [68] or Vecchia-approximated GPs [69], or by employing a parameter dimension reduction technique (e.g. PCA) prior to performing emulation and inference [11].

9. Conclusions

In this study, several state-of-the-art emulation methods based on Gaussian Processes and deep residual neural networks were employed and assessed to emulate blood flow and pressure in the pulmonary and systemic blood circulation using simulations from two computational fluid-dynamics models of the pulmonary and systemic system. Emulation has been used as a computationally efficient tool to perform inference of unknown model parameters using optimisation techniques, and MCMC, History Matching and rejection sampling approaches, which are established inference techniques, widely used in the machine learning community. Moreover, robust assessment criteria have been developed to compare the performance of the proposed methods with respect to accuracy and UQ. An essential part of the assessment process is that the parameter estimation and UQ analysis needed to be conducted within a limited time interval to provide real-time model calibration. The aim is to develop inference techniques that can be employed for decision support in a clinical setting to enable personalised medicine. We find that for the systemic challenge, which featured an idealised case of noise-free data, the relative deviation from the ground-truth in parameter space ranges from 10^{-5}% (highest-performing method) to 3% (lowest-performing method). For the pulmonary challenge, for which noisy data were generated, the relative deviation ranges from 0.9% to 7% for the parameter posterior mean, and from 35% to 570% for the parameter posterior variance.

CRediT authorship contribution statement

L. Mihaela Paun: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Mitchel J. Colebank:** Writing – review & editing, Writing – original draft, Software, Data curation. **Alyssa Taylor-LaPole:** Writing – review & editing, Writing – original draft, Software, Data curation. **Mette S. Olufsen:** Writing – review & editing, Writing – original draft, Supervision, Software. **William Ryan:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Iain Murray:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **James M. Salter:** Writing – review & editing, Writing – original

draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Victor Applebaum:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Michael Dunne:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Jake Hollins:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Louise Kimpton:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Victoria Volodina:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Xiaoyu Xiong:** Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Dirk Husmeier:** Writing – review & editing, Writing – original draft, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

The code and data are available at https://github.com/LMihaelaPaun/SECRET.git.

Acknowledgements

The competition was organised as part of the research programme of the SofTMech Statistical Emulation and Translation Hub, funded by EPSRC, grant reference number EP/T017899/1. We thank Dr Gillian Brown for the tremendous help with the competition organisation. We also acknowledge the following: National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant TL1 TR001415 (Colebank); National Institutes of Health (NIH) National Heart, Lung, and Blood Institute (NHLBI) grant HL154624 (Colebank); National Science Foundation Graduate Research Fellowship under Grant No. DGE-2137100 (Taylor-LaPole); NIH NHLBI grant R01 HL147590 (Olufsen). For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2024.117193.

References

[1] M.E.J. Newman, Resource letter CS–1: Complex systems, Am. J. Phys. 79 (8) (2011) 800–810.
[2] M. Banwarth-Kuhn, S. Sindi, How and why to build a mathematical model: A case study using prion aggregation, J. Biol. Chem. 295 (2020) jbc.REV119.009851.
[3] D. Raje, S. Ghosh, P.P. Mujumdar, Hydrologic impacts of climate change: Quantification of uncertainties, in: Climate Change Modeling, Mitigation, and Adaptation, 2013, pp. 177–218.
[4] N. Perra, B. Gonçalves, Modeling and predicting human infectious diseases, Soc. Phenomena 23 (2015) 59–83.
[5] A. Corti, A. McQueen, F. Migliavacca, C. Chiastra, S. McGinty, Investigating the effect of drug release on in-stent restenosis: A hybrid continuum – agent-based modelling approach, Comput. Methods Programs Biomed. 241 (2023) 107739.
[6] J. Zambrano-Martinez, C. Calafate, D. Soler, J.-C. Cano, P. Manzoni, Modeling and characterization of traffic flows in urban environments, Sensors 18 (2018) 2020.
[7] C. Gilbert, J. Browell, D. McMillan, Leveraging turbine-level data for improved probabilistic wind power forecasting, IEEE Trans. Sustain. Energy 11 (3) (2020) 1152–1160.
[8] A. Haleem, M. Javaid, R. Pratap Singh, R. Suman, Exploring the revolution in healthcare systems through the applications of digital twin technology, Biomed. Technol. 4 (2023) 28–38.
[9] F. Mohsen, B. Al-Saadi, N. Abdi, S. Khan, Z. Shah, Artificial intelligence-based methods for precision cardiovascular medicine, J. Personalized Med. 13 (8) (2023).
[10] M.U. Qureshi, G.D.A. Vaughan, C. Sainsbury, M. Johnson, C.S. Peskin, M.S. Olufsen, N.A. Hill, Numerical simulation of blood flow and pressure drop in the pulmonary arterial and venous circulation, Biomech. Model. Mechanobiol. 13 (2014) 1137–1154.
[11] A. Lazarus, H. Gao, X. Luo, D. Husmeier, Improving cardio-mechanic inference by combining in vivo strain data with ex vivo volume–pressure data, J. R. Stat. Soc. Ser. C. Appl. Stat. 71 (4) (2022) 906–931.
[12] L.M. Paun, M.J. Colebank, M.S. Olufsen, N.A. Hill, D. Husmeier, Assessing model mismatch and model selection in a Bayesian uncertainty quantification analysis of a fluid-dynamics model of pulmonary blood circulation, J. R. Soc. Interface 17 (173) (2020) 20200886.
[13] S. Conti, A. O'Hagan, Bayesian emulation of complex multi-output and dynamic computer models, J. Statist. Plann. Inference 140 (3) (2010) 640–651.
[14] T. Bharucha, M. Hlavacek, D.E. Spicer, P. Theocharis, R.H. Anderson, How should we diagnose and differentiate hearts with double-outlet right ventricle? Cardiol. Young 27 (1) (2017) 1–15.
[15] A.M. Taylor-LaPole, M.J. Colebank, J.D. Weigand, M.S. Olufsen, C. Puelz, A computational study of aortic reconstruction in single ventricle patients, Biomech. Model. Mechanobiol. 22 (2023) 357–377.
[16] D. Navaratnam, S. Fitzsimmons, M. Grocott, H.B. Rossiter, Y. Emmanuel, G. Diller, T. Gordon-Walker, S. Jack, N. Sheron, J. Pappachan, J.N. Pratap, J.J. Vettukattil, V. G, Exercise-induced systemic venous hypertension in the fontan circulation, Am. J. Cardio. 117 (2016) 1667–1671.
[17] T.T. Gordon-Walker, K. Bove, G. Veldtman, Fontan-associate liver disease: A review, J. Cardiol. 74 (2019) 223–232.

[18] M.S. Olufsen, C.S. Peskin, W.Y. Kim, E.M. Pedersen, A. Nadim, J. Larsen, Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions, Ann. Biomed. Eng. 28 (2000) 1281–1299.
[19] S.J. Roberts, M.A. Osborne, M. Ebden, S. Reece, N.P. Gibson, S. Aigrain, Gaussian processes for time-series modelling, Phil. Trans. R. Soc. A 371 (2013).
[20] M. Gu, X. Wang, J.O. Berger, Robust Gaussian stochastic process emulation, Ann. Statist. 46 (6A) (2018) 3038–3066.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015.
[22] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, Comput. J. 7 (2) (1964) 155–162.
[23] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[24] S. Brooks, A. Gelman, G. Jones, X.-L. Meng (Eds.), Handbook of Markov Chain Monte Carlo, Chapman & Hall / CRC Press, 2011.
[25] P.S. Craig, M. Goldstein, A.H. Seheult, J.A. Smith, Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments, in: Case Studies in Bayesian Statistics: Volume III, Springer, 1997, pp. 37–93.
[26] B. Moskowitz, R. Caflisch, Smoothness and dimension reduction in quasi-Monte Carlo methods, Math. Comput. Modelling 23 (8–9) (1996) 37–54.
[27] J. Ryu, X. Hu, S.C. Shadden, A coupled lumped-parameter and distributed network model for cerebral pulse-wave hemodynamics, J. Biomech. Eng. 137 (10) (2015) 101009.
[28] P. Lax, B. Wendroff, Systems of conservation laws, Commun. Pure Appl. Math. 13 (2) (1960) 217–237.
[29] T. Tabata, J.D. Thomas, A.L. Klein, Pulmonary venous flow by doppler echocardiography: revisited 12 years later, J. Am. College Cardiol. 41 (2003) 1243–1250.
[30] J.P. Mynard, J.J. Smolich, One-dimensional haemodynamic modeling and wave dynamics in the entire adult circulation, Ann. Biomed. Eng. 43 (2015) 1443–1460.
[31] M.U. Qureshi, M.J. Colebank, D.A. Schreier, D.M. Tabima, M.A. Haider, N.C. Chesler, M.S. Olufsen, Characteristic impedance: frequency or time domain approach? Physiol. Meas. 39 (2018) 014004.
[32] C.K. Williams, C.E. Rasmussen, Gaussian Processes for Machine Learning, vol. 2, (3) MIT press Cambridge, MA, 2006.
[33] M.C. Kennedy, A. O'Hagan, Bayesian calibration of computer models, J. R. Stat. Soc. Ser. B Stat. Methodol. 63 (3) (2001) 425–464.
[34] D. Higdon, J. Gattiker, B. Williams, M. Rightley, Computer model calibration using high-dimensional output, J. Amer. Statist. Assoc. 103 (482) (2008) 570–583.
[35] D.M. Sexton, J.M. Murphy, M. Collins, M.J. Webb, Multivariate probabilistic projections using imperfect climate models part I: outline of methodology, Clim. Dyn. 38 (11–12) (2011) 2513–2542.
[36] W. Chang, M. Haran, P. Applegate, D. Pollard, Calibrating an ice sheet model using high-dimensional binary spatial data, J. Amer. Statist. Assoc. 111 (513) (2016) 57–72.
[37] J.M. Salter, D.B. Williamson, J. Scinocca, V. Kharin, Uncertainty quantification for computer models with spatial output using calibration-optimal bases, J. Amer. Statist. Assoc. 114 (528) (2019) 1800–1814, arXiv:1801.08184.
[38] S. Coveney, C. Corrado, J.E. Oakley, R.D. Wilkinson, S.A. Niederer, R.H. Clayton, Bayesian calibration of electrophysiology models using restitution curve emulators, Front. Physiol. (2021) 1120.
[39] L.S. Bastos, A. O'Hagan, Diagnostics for Gaussian process emulators, Technometrics 51 (4) (2009) 425–438.
[40] P.S. Craig, M. Goldstein, A.H. Seheult, J.A. Smith, Bayes linear strategies for matching hydrocarbon reservoir history, in: Bayesian Statistics 5: Proceedings of the Fifth Valencia International Meeting, Oxford University Press, 1996.
[41] O. Tange, GNU Parallel 2018, Ole Tange, 2018.
[42] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: KDD 2016: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[43] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: J.C. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems 20, MIT Press, Cambridge, MA, 2008, pp. 1177–1184.
[44] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, 2015.
[45] P. Kidger, C. Garcia, Equinox: neural networks in JAX via callable PyTrees and filtered transformations, 2021, Differentiable Programming workshop at Neural Information Processing Systems 2021.
[46] L. Metz, J. Harrison, C.D. Freeman, A. Merchant, L. Beyer, J. Bradbury, N. Agrawal, B. Poole, I. Mordatch, A. Roberts, J. Sohl-Dickstein, VeLO: training versatile learned optimizers by scaling up, 2022, Preprint arXiv:2211.09760.
[47] D. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference for Learning Representations, ICLR, 2015, arXiv:1412.6980.
[48] K. Pearson, On lines and planes of closest fit to systems of points in space, Lond. Edinb. Dublin Philos. Mag. J. Sci. 2 (11) (1901) 559–572.
[49] D. Foreman-Mackey, D.W. Hogg, D. Lang, J. Goodman, Emcee: The MCMC hammer, Publ. Astron. Soc. Pac. 125 (925) (2013) 306.
[50] F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, Comput. Optim. Appl. 51 (2012) 259–277.
[51] H. Haario, E. Saksman, J. Tamminen, An adaptive Metropolis algorithm, Bernoulli 7 (2) (2001) 223–242.
[52] A.W. Bowman, A. Azzalini, Applied Smoothing Techniques for Data Analysis, Oxford University Press Inc., New York, 1997.
[53] B. Silverman, Density Estimation for Statistics and Data Analysis, in: London: Chapman & Hall/CRC, Chapman and Hall, 1986.
[54] D.W. Scott, G.R. Terrell, Biased and unbiased cross-validation in density estimation, J. Amer. Statist. Assoc. 82 (400) (1987) 1131–1146.
[55] I. Murray, D. MacKay, R.P. Adams, The Gaussian process density sampler, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems, Vol. 21, Curran Associates, Inc., 2008.
[56] J. Riihimäki, A. Vehtari, Laplace approximation for logistic Gaussian process density estimation and regression, Bayesian Anal. 9 (2012) 425–448.
[57] H. Jeffreys, An invariant form for the prior probability in estimation problems, Proc. R. Soc. Lond. A 186 (1007) (1946) 453–461.
[58] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (25) (2012) 723–773.
[59] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[60] J.M. Salter, D. Williamson, A comparison of statistical emulation methodologies for multi-wave calibration of environmental models, Environmetrics 27 (8) (2016) 507–523.
[61] L.M. Paun, M. Colebank, M. Umar Qureshi, M. Olufsen, N. Hill, D. Husmeier, MCMC with Delayed Acceptance using a Surrogate Model with an Application to Cardiovascular Fluid Dynamics, in: Proceedings of the International Conference on Statistics: Theory and Applications, ICSTA'19, 2019.
[62] L. Paun, D. Husmeier, Emulation-accelerated Hamiltonian Monte Carlo algorithms for parameter estimation and uncertainty quantification in differential equation models, Stat. Comput. 32 (1) (2022).
[63] J. Brynjarsdóttir, A. O'Hagan, Learning about physical parameters: The importance of model discrepancy, Inverse Problems 30 (11) (2014) 114007.
[64] D. Tran, R. Ranganath, D. Blei, The variational Gaussian process, in: 4th International Conference on Learning Representations, ICLR 2016, 2016.
[65] A. López-Lopera, N. Durrande, M. Álvarez, Physically-inspired Gaussian process models for post-transcriptional regulation in drosophila, IEEE/ACM Trans. Comput. Biol. Bioinform. 18 (2) (2021) 656–666.
[66] C.M. Fleeter, G. Geraci, D.E. Schiavazzi, A.M. Kahn, A.L. Marsden, Multilevel and multifidelity uncertainty quantification for cardiovascular hemodynamics, Comput. Methods Appl. Mech. Engrg. 365 (2020) 113030.
[67] R.B. Gramacy, D.W. Apley, Local Gaussian process approximation for large computer experiments, J. Comput. Graph. Statist. 24 (2) (2015) 561–578.
[68] V. Dutordoir, N. Durrande, J. Hensman, Sparse Gaussian processes with spherical harmonic features, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, Vol. 119, PMLR, 2020, pp. 2793–2802.
[69] A.C. Annie Sauer, R.B. Gramacy, Vecchia-approximated deep Gaussian processes for computer experiments, J. Comput. Graph. Statist. 32 (3) (2023) 824–837.