Diploma Thesis

Uncertainty Quantification of Deep Learning Reduced-Order Models for Computational Fluid Dynamics

submitted in satisfaction of the requirements for the degree

Diplom-Ingenieur

of the TU Wien, Faculty for Mechanical and Industrial Engineering

---

Diplomarbeit

Unsicherheitsquantifizierung von Deep-Learning Reduced-Order-Modellen für numerische Strömungsmechanik

ausgeführt zum Zwecke der Erlangung des akademischen Grads

Diplom-Ingenieur

eingereicht an der TU Wien, Fakultät für Maschinenwesen und Betriebswissenschaften

Gabriel Sperrer, BSc

Matr.Nr.: 11813052

Betreuung: Univ.Prof. Dr.-Ing. **Manuel Garcia-Villalba Navaridas**
Institut für Strömungsmechanik und Wärmeübertragung
Forschungsbereich Numerische Strömungsmechanik
Technische Universität Wien
Prof. dr. ir. **Wouter Huberts**
Biomedical Engineering, Cardiovascular Biomechanics,
Technische Universität Eindhoven
Institut für Informatik, Computational Science Lab,
Universiteit van Amsterdam

Wien, im Oktober 2025

Kurzfassung

Die numerische Strömungsmechanik (Computational Fluid Dynamics, CFD) wird in zahlreichen wissenschaftlichen und klinischen Anwendungen zur Vorhersage von Strömungs- und Druckfeldern eingesetzt. Für kritische Problemstellungen hängt die Verlässlichkeit solcher Vorhersagen jedoch nicht nur von deren Genauigkeit, sondern auch von der Sicherheit bzw. Unsicherheit im Zusammenhang mit der Vorhersage ab. Diese Arbeit präsentiert ein Modell zur Unsicherheitsquantifizierung in Deep-Learning-basierten Reduced-Order-Modellen für CFD-Simulationen.

Aufbauend auf der von Fresca et al. eingeführten POD-DL-ROM-Architektur wird diese um einen heteroskedastischen Ausgang erweitert, um aleatorische Unsicherheit zu erfassen, und dann als ensemble Trainiert, um epistemische Unsicherheit abzuschätzen. Diese Erweiterungen ermöglichen die Zerlegung der Varianz in aleatroische und epistemische Anteile.

Das vorgeschlagene Modell wird anhand von drei Benchmark-Problemen zunehmender Komplexität validiert: einem instationären Advektions-Diffusions-Reaktions-System, der laminaren von-Kármán-Wirbelstraße und der Blutströmung in einem dreidimensionalen zerebralen Aneurysma. In allen Fällen erreicht das Ensemble-POD-DL-ROM eine Genauigkeit, die mit dem Stand der Technik vergleichbar oder besser ist, und liefert Unsicherheitsabschätzungen, die inhaltlich mit den Modellfehlern korrelieren.

Die wichtigsten Beiträge dieser Arbeit sind (i) die Erweiterung der POD-DL-ROM Architektur zur Vorhersage von Unsicherheit, (ii) der Nachweis der Anwendbarkeit des Ansatzes auf zwei typische Benchmark-Szenarien sowie ein klinisch relevantes hämodynamisches Beispiel sowie (iii) eine systematische Analyse des Zusammenhangs zwischen der vorhergesagten Unsicherheit und dem Fehler in den Vorhersagen. Die Ergebnisse zeigen, dass kompakte Ensembles von POD-DL-ROMs schnelle, genaue und unsicherheitsbewusste Strömungsvorhersagen ermöglichen und damit eine Grundlage für vertrauenswürdige Reduced-Order-Modelle in der numerischen Strömungsmechanik schaffen.

Abstract

Computational fluid dynamics (CFD) is widely used to predict flow and pressure fields in scientific and clinical applications. However, for decision-critical tasks, the reliability of these predictions depends not only on their accuracy but also on well-quantified uncertainty. This thesis presents a framework for uncertainty quantification in deep-learning reduced-order models for CFD.

We extend the POD-DL-ROM architecture introduced by Fresca et al. by equipping it with a heteroscedastic output to capture aleatoric uncertainty and by training an ensemble of networks to estimate epistemic uncertainty. Together, these modifications enable the decomposition of total predictive variance into data- and model-based components within a compact and computationally tractable surrogate.

The framework is validated on three benchmark problems of increasing complexity: an unsteady advection–diffusion–reaction system, the laminar von Kármán vortex street, and blood flow in a three-dimensional cerebral aneurysm. Across all cases, the proposed ensemble POD-DL-ROM achieves accuracy comparable to or better than state-of-the-art reduced-order models while providing uncertainty estimates that meaningfully correlate with model error.

This work contributes (i) the extension of the POD-DL-ROM to enable predicting uncertainty, (ii) a demonstration of the model's applicability to two classic benchmark cases in CFD as well as a relevant clinical example, and (iii) a systematic analysis of the relation between predicted uncertainty and model error. The results show that compact ensembles of POD-DL-ROMs can deliver fast, accurate, and uncertainty-aware flow field predictions, establishing a foundation for trustworthy reduced-order modeling in computational fluid dynamics.

Contents

Chapter 1

Introduction

Fluid dynamics is a scientific discipline with a long history in many fields of science and engineering. Being based on the Navier–Stokes equations, partial differential equations (PDEs), which are hard to solve analytically, it took until the advent of modern computing and numerical solvers for fluid dynamics to become a practical tool for solving real-world problems [1]. The development of computational fluid dynamics (CFD) has enabled the numerical solution of fluid dynamics problems, allowing for the simulation of complex fluid flows [2]. This has led to significant advancements in fields such as aerospace engineering [3], structural engineering [4], and biomedical engineering [5]. Methods such as the finite difference method (FDM), the finite element method (FEM), and finite volume method (FVM) are among the standard methods for solving fluid dynamics problems. However, these high-fidelity methods (also called full-order models) require a sufficiently fine mesh to resolve complex flow patterns, ultimately yielding large-scale nonlinear systems of equations to be solved. For tasks requiring fast execution time, such as problems dealing with real-time predictions or multi-query tasks, the computational demands of the solvers for these equations make them infeasible to use. In the former case, the solution must be computed in a very limited amount of time, whereas in the latter, many solutions for different input parameter combinations must be computed at once.

1.1 Related work

There has therefore been a lot of work dedicated to replacing high-dimensional full-order models (FOMs) with surrogate models of reduced order, also called reduced-order models (ROMs), that approximate the FOM solution at a fraction of the computational cost [6], [7]. The basic assumption underlying many ROMs is that the solution of the FOM lies on a low-dimensional manifold, meaning that the solution often lies on a smooth subspace of the full solution space. This can often be exploited to reduce the computational complexity by performing calculations in the reduced space [8]. A prominent family of ROMs are projection-based methods, which project the discretized solution of the full-order PDEs onto a reduced subspace spanned by basis functions. Examples include the proper orthogonal decomposition (POD), which constructs a reduced basis by exploiting the singular value decomposition (SVD) of a snapshot matrix of FOM solutions. When combined with Galerkin projection of the governing equations onto this reduced basis, one obtains the classical POD–Galerkin ROM, which has long been the standard projection-based approach in CFD [6], [7].

While powerful, projection-based ROMs face two main limitations. First, the restriction to a linear subspace can limit their ability to capture complex nonlinear solutions, as often encountered in fluid dynamics. For problems with large parametric variations or strong dependence of coherent structures on parameters, the dimension of the required linear subspace can become very large, yielding diminishing returns in speed-ups as well as potential stability issues. Second, even when the reduced dimension is moderate, the evaluation of nonlinear operators in the projected equations still depends on the full-order dimension [9]. To address this, hyper-reduction techniques are employed to reduce the complexity [9]. These methods, however, do not alleviate the fundamental limitation of linear subspaces in representing nonlinear solution manifolds.

As an alternative approach, there has been growing interest in deep-learning (DL) surrogate models. The past decade has witnessed tremendous progress in deep learning across domains such as computer vision and natural language processing, and more recently in scientific computing, with breakthroughs in drug discovery [10], protein folding [11], and antibiotics research [12]. The key advantage of DL methods lies in their ability to approximate highly nonlinear function mappings directly from data, making them particularly well suited to learn the complex mappings characterizing fluid dynamics problems. In contrast to projection-based ROMs, which reduce the governing equations of the FOM via projection, DL-ROMs are typically non-intrusive and learn a data-driven mapping from problem parameters to the PDE solution manifold.

In recent years, different deep learning architectures have been proposed in the realm of scientific computing and specifically CFD, differing in how they incorporate physical knowledge and data into surrogate models [13].

One influential line of research is Physics-Informed Neural Networks (PINNs) [14], where the governing PDEs are encoded into the loss function of the neural network. This allows PINNs to integrate limited or scattered observational data with strong physical constraints, making them particularly well suited for inverse problems such as parameter identification, data assimilation, or model calibration. Their ability to enforce physical laws without requiring dense training datasets has made them attractive in physics and biomedical applications where measurements are scarce [15].

Another promising direction is that of Neural Operators, such as the Deep Operator Network (DeepONet) [16] and the Fourier Neural Operator (FNO) [17]. These methods aim to learn mappings between infinite-dimensional function spaces, directly approximating the operator that maps PDE inputs (e.g., boundary conditions or forcing terms) to solutions. Neural Operators have demonstrated strong generalization across parameter ranges and even across different discretizations, making them powerful tools for operator learning.

In parallel, hybrid approaches such as the POD–DL–ROM framework [18] combine classical reduced-order modeling with deep learning. The POD-DL-ROM framework is characterized by the idea of first obtaining a reduced basis from snapshots using proper orthogonal decomposition, and then in addition learning a nonlinear mapping in the reduced space using neural networks. In contrast to the aforementioned approaches, applying linear dimensionality reduction before training a neural network reduces both the input and output dimensionality of the neural network, which makes training considerably more efficient and less data-intensive. This compactness is particularly advantageous in settings that demand the training of multiple models in parallel, such as when training ensembles of models. While ensemble deep learning methods have originally been introduced to increase model accuracy [19], [20], they also provide a practical means to quantify model uncertainty [21], [22].

Uncertainty quantification (UQ) is an important aspect of all model building and especially so in environments where models influence high-stakes decisions, such as autonomous driving [23], [24], healthcare [25]–[27], and medicine [28]. Models trained on data exhibit multiple sources of uncertainty, which are commonly categorized into two classes: aleatoric uncertainty, attributable to inherent randomness or noise in the data, and epistemic uncertainty, corresponding to the limited knowledge of the model or its parameters [23]. In the realm of scientific computing, UQ has received significant attention with vast literature covering the topic and a variety of methods available [29]. Aleatoric uncertainty can be assumed homoscedastic, meaning that the variance is constant for all outputs, or heterosedastic, meaning that it varies dependent on the inputs. The latter, more complex case, has been successfully modeled explicitly, for example, by enabling a model to learn the variance from the data [23]. Building on this insight, assumed density filtering was proposed [30], which learns the aleatoric uncertainty not only based on the target data, but also by propagating input uncertainty. When it comes to epistemic uncertainty regarding deep learning models in particular, most UQ approaches are grounded in Bayesian formalism [31]. To estimate epistemic uncertainty, one approach is to specify a prior distribution over the network's parameters. The posterior for those parameters is then approximated using the training data, since, due to the nonlinearity and size of neural networks, exact Bayesian inference is intractable. Bayesian neural networks (BNNs) [32] perform this approximation through sampling-based approaches such as Markov Chain Monte Carlo (MCMC) [32], [33] or variational inference [34], [35]. However, due to computational challenges and implementation complexity, Bayesian NNs often remain slower and harder to deploy than standard neural networks. To overcome these limitations, Monte Carlo Dropout (MC Dropout) was proposed [36], which interprets the use of dropout layers [37] as approximate Bayesian inference. At test time, dropout is kept active to generate a distribution over predictions with minimal modifications to the network architecture. While its simplicity has led to widespread adoption, MC Dropout can underestimate predictive uncertainty and its performance depends strongly on hyperparameters such as dropout rate and weight decay, which implicitly correspond to prior choices [36], [38]. Moreover, dropout may act as a regularizer and thereby reduce predictive accuracy [39]. Motivated by the interpretation of dropout as training implicit ensembles, Deep Ensembles were introduced as a generalized and empirically more robust approach [21].

Ensemble methods train multiple models independently. While they are most commonly used to improve prediction in a variety of machine learning tasks not limited to neural networks [40], they can also be used to obtain a robust estimate of uncertainty [21]. The diversity among ensemble members can be achieved through different means, such as random initializations and data shuffles. Ensembles have been shown to provide well-calibrated uncertainty estimates and to be more robust to overfitting compared to single models [39]. However, training multiple models can be computationally expensive, especially for large networks. Therefore, ensemble methods are most practical when the individual models are relatively compact.

1.2 Contributions

In this thesis, we show how uncertainty in CFD solutions can be quantified in a systematic and interpretable way. The main contributions are:

- A framework for UQ of CFD simulations: We train Deep Ensembles of the POD-DL-ROM to quantify predictive uncertainty in CFD solutions. The total uncertainty is decomposed into aleatoric and epistemic uncertainty. Aleatoric uncertainty is learned from the data using a heteroscedastic model, while epistemic uncertainty is obtained from variance across ensemble members.

- Application to benchmark CFD problems: We demonstrate the generality of the proposed framework by applying it to a range of standard benchmark problems. In all cases, we show that the proposed framework is able to achieve accurate mean-field predictions while providing calibrated uncertainty estimates.

- Assessment of uncertainty quality: Beyond quantifying the magnitude of predictive uncertainty, we evaluate its quality using calibration (CE) and ranking (AUSE) metrics, adapted from probabilistic regression. This allows assessing whether the predicted variances are statistically consistent with observed errors.

1.3 Structure of the Thesis

The rest of the thesis is structured as follows. Chapter 2 provides the necessary theoretical foundations for our model, introducing reduced-order modeling, the foundations of deep learning, and uncertainty quantification. In Chapter 3, we assess the performance of the model three different test cases, namely (i) a linear advection-diffusion-reaction problem, (ii) the von Kármán vortex street, and (iii) the flow in a cerebral aneurysm. Finally, Chapter 4 concludes the work by discussing the results, summarizing the main findings of this work, and providing directions for future research.

Chapter 2

Uncertainty Quantification of CFD Simulations

This chapter provides the methodological background required for the development of deep learning–based reduced-order models with uncertainty quantification. We first review projection-based reduced-order modeling techniques, with emphasis on the Proper Orthogonal Decomposition, establishing the notation used throughout the thesis. We then introduce the essential building blocks of deep learning relevant for surrogate modeling of CFD problems, in particular multi-layer perceptrons, convolutional neural networks, and autoencoders. This subsection ends with the presentation of the POD-DL-ROM, a POD-enhanced deep learning model that treats POD as the first step in a reduced-order model based on deep learning. Finally, we present the fundamentals of uncertainty quantification for deep learning, with a special focus on the estimation of aleatoric and epistemic uncertainty. Together, these components provide the methodological basis for constructing deep learning reduced-order models with quantified uncertainty, which we apply in the subsequent chapters.

For the purposes of this thesis, we focus on incompressible fluid dynamics problems with rigid walls on a fixed domain $\Omega \subset \mathbb{R}^d$ with $d = 2, 3$, complemented by suitable initial and boundary conditions. See the examples in Chapter 3 for specific problem settings. The spatial discretization of the governing equations by finite elements or finite volumes yields a nonlinear dynamical system of dimension $N_h \gg 1$, which we denote as the full–order model (FOM). After time discretization with $\Delta t = T/N_t$, one obtains a sequence of algebraic systems to be solved at each time step $t_k = k\Delta t$, $k = 1, \ldots, N_t$. The dimension N_h corresponds to the number of cells used for the meshes in each case.

2.1 Reduced Order Modeling

The goal of reduced order modeling is to replace a high-fidelity or full-order model with a reduced-order model, which is capable of expressing the same dynamics as the full model, but at a fraction of the computational cost. Among the different approaches, reduced basis (RB) methods remain one of the most popular options [8]. The assumption behind these methods is that the solution of a high-dimensional parametrized PDE lies in a low-dimensional manifold [6], [7]. A reduced basis method typically consists of two stages: an offline stage, where the reduced basis is constructed, and an online stage, where the reduced basis is used to solve the parametrized PDE for new parameter values. The computationally expensive offline stage is used to construct the low-dimensional manifold and assemble the ROM arrays, whereas the online stage can then be used to compute solution approximations for new parameter combinations in a computationally efficient manner.

RB methods approximate the FOM solution in low-dimensional subspaces spanned by a set of basis vectors. Hence, a linear ROM approximates the high-fidelity solution u_h for any represents a field of interest such as velocity or pressure in the form

$$\tilde{u}_h(t;\mu) = V_N \, u_N(t;\mu), \tag{2.1}$$

where $V_N \in \mathbb{R}^{N_h \times N}$ is the reduced basis matrix containing the basis vectors, and $u_N(t;\mu) \in \mathbb{R}^N$ are the projected coordinates from the high-dimensional solution space of the FOM $\mathbb{R}^{N_h}$ onto the new subspace $\mathbb{R}^N$ spanned by the basis vectors. V_N is computed from snapshot data using Proper Orthogonal Decomposition (POD). Given N_{train} training parameter values and N_t time instances, we collect the corresponding high-fidelity solutions into the snapshot matrix

$$S_u = [u_h(t^1;\mu_1) \mid \ldots \mid u_h(t^{N_t};\mu_{N_{\text{train}}})] \in \mathbb{R}^{N_h \times N_t N_{\text{train}}}. \tag{2.2}$$

POD identifies an orthonormal basis representing a set of snapshots, which is mathematically equivalently to the Singular Value Decomposition (SVD). For any given snapshot matrix S_u,

$$S_u = U \Sigma V^\top, \tag{2.3}$$

where the left singular vectors U are the POD modes, the singular values Σ determine their relative energetic importance, and the right singular vectors V are the temporal coefficients.

Since both N_h and $N_s = N_t N_{\text{train}}$ may be very large, we employ randomized singular value decomposition (rSVD) [41] to efficiently approximate the first N singular vectors. The reduced basis is achieved by truncating after the N singular vectors with the highest amount of energy, therefore describing the biggest part of the variance of the system. The truncation index N is selected so that the retained error is below a prescribed tolerance τ.

2.2 Deep Learning for Reduced Order Modeling

In contrast to projection-based ROMs, data-driven reduced-order models do not derive reduced dynamics from the governing equations, but instead learn a direct mapping from input parameters to the solution field using data. This circumvents the limitations of linear trial subspaces, enabling efficient approximation of nonlinear trial manifolds.

A neural network processes input features $x \in \mathbb{R}^n$ through a series of nonlinear transformations,

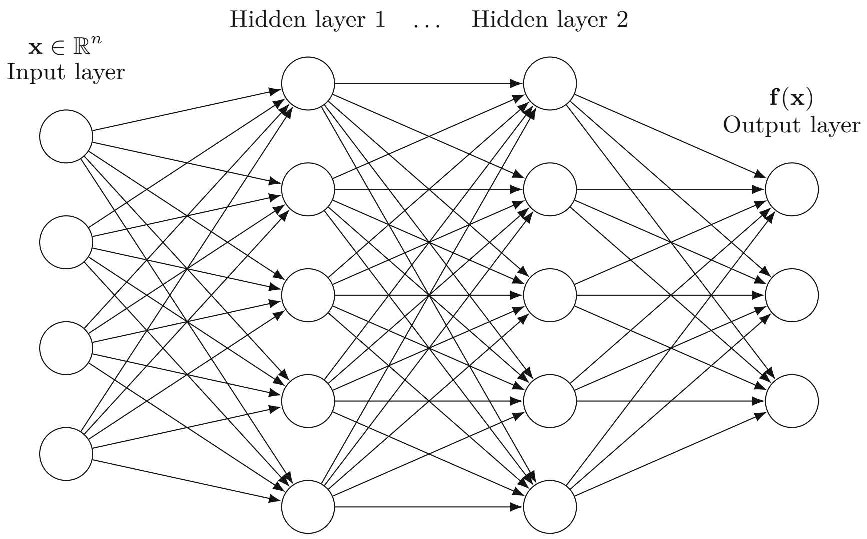
$$f(x) = \sigma(Wx + b), \tag{2.4}$$

where W and b denote weights and biases, and σ is a nonlinear activation function, with the rectified linear unit (ReLU) and its variants being common choices [42]. The arrangement of neurons and layers defines the network architecture. Among the most common are multi-layer perceptrons (also known as fully connected feedforward networks), convolutional neural networks, and recurrent neural networks. In this work we focus on feedforward networks and CNNs.

Multi-layer perceptrons (MLPs) are composed of layers of neurons, where each layer applies an affine transformation followed by a nonlinear activation. Collectively, the network defines a nonlinear mapping

$$f(x) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(x), \tag{2.5}$$

where L denotes the number of layers. MLPs of an input layer, one or more hidden layers, and an output layer, where each layer is given by eq 2.4. Despite their architectural simplicity, they are universal function approximators, capable of approximating any continuous function under mild assumptions [43], [44].



$$\mathbf{f}(\mathbf{x}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(\mathbf{x}), \quad f^{(l)}(\mathbf{x}) = \sigma\big(W^{(l)}\mathbf{x} + \mathbf{b}^{(l)}\big)$$

Fig. 2.1: Multi-layer perceptron with fully connected layers. Each layer applies an affine map $W^{(l)}(\cdot) + \mathbf{b}^{(l)}$ followed by a nonlinearity σ, yielding the composition $\mathbf{f}(\mathbf{x}) = f^{(L)} \circ \cdots \circ f^{(1)}(\mathbf{x})$.

Convolutional neural networks (CNNs) extend feedforward networks by including convolutional layers [45]. A two-dimensional convolutional layer applies a set of learnable kernels (or filters) $K \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times k_x \times k_y}$ to an input feature map $X \in \mathbb{R}^{C_{\text{in}} \times W_{\text{in}} \times H_{\text{in}}}$, producing an output feature map $Y \in \mathbb{R}^{C_{\text{out}} \times W_{\text{out}} \times H_{\text{out}}}$. Each output channel Y_c is obtained by convolving the corresponding kernel with all input channels and summing the results:

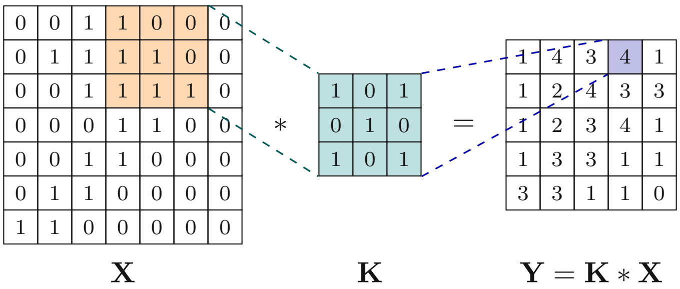
$$Y_c = \sum_{i=1}^{C_{\text{in}}} K_{c,i} * X_i, \qquad c = 1, \dots, C_{\text{out}}, \tag{2.6}$$

where $*$ denotes the 2D convolution. Nonlinear activations are typically applied after each convolution. By exploiting local connectivity, where each neuron is only connected to a local region of the input, and weight sharing, where the same learnable kernel is applied across the input field, CNNs capture spatial correlations with far fewer parameters than fully connected layers. This makes them particularly effective for high-dimensional fields such as velocity or pressure. Consequently, CNNs have become a popular choice for learning fluid dynamics from data [46], [47].

Neural networks can be trained using different learning paradigms. In supervised learning, the network is provided with pairs of inputs and target outputs and is trained to approximate the input–output relation. In unsupervised learning, the network is instead trained to identify structure in the input data without explicit targets. Autoencoders, which will be introduced in the next subsection, are a common form of unsupervised learning model.

2.2.1 Autoencoders

An autoencoder (AE) is a neural network trained to reconstruct its input through a low-dimensional bottleneck [48], [49]. It consists of an encoder $E_\theta : \mathbb{R}^N \to \mathbb{R}^n$ and a decoder $D_\phi : \mathbb{R}^n \to \mathbb{R}^N$ with $n \ll N$:



$$\mathbf{X} = \begin{pmatrix} 0&0&1&1&0&0&0\\ 0&1&1&1&1&0&0\\ 0&0&1&1&1&1&0\\ 0&0&0&1&1&0&0\\ 0&0&1&1&0&0&0\\ 0&1&1&0&0&0&0\\ 1&1&0&0&0&0&0 \end{pmatrix} * \mathbf{K} = \begin{pmatrix} 1&0&1\\ 0&1&0\\ 1&0&1 \end{pmatrix} = \mathbf{Y} = \begin{pmatrix} 1&4&3&4&1\\ 1&2&4&3&3\\ 1&2&3&4&1\\ 1&3&3&1&1\\ 3&3&1&1&0 \end{pmatrix}$$

Fig. 2.2: Illustration of a 2D convolution. A 3 × 3 kernel $\mathbf{K}$ slides over the input $\mathbf{X}$ (valid padding) to produce the output $\mathbf{Y} = \mathbf{K} * \mathbf{X}$. Highlighted cells show one receptive field and its mapped output entry, matching the notation in the text.

$$z = E_\theta(x), \qquad \tilde{x} = D_\phi(z), \tag{2.7}$$

where $x \in \mathbb{R}^N$ is the input, $z \in \mathbb{R}^n$ the latent code, and $\tilde{x}$ the reconstruction. The parameters (θ, ϕ) are learned by minimizing a reconstruction loss over snapshots $\{x^{(j)}\}_{j=1}^{N_s}$:

$$\min_{\theta,\phi} \frac{1}{N_s} \sum_{j=1}^{N_s} \|x^{(j)} - D_\phi(E_\theta(x^{(j)}))\|_2^2. \tag{2.8}$$

AEs perform nonlinear dimensionality reduction: the decoder D_ϕ defines a nonlinear trial manifold $\mathcal{M} = \{D_\phi(z) \mid z \in \mathbb{R}^n\}$ that approximates the data. In the special case of a single hidden layer with linear activations, mean-centered inputs, and squared-error loss, the optimal encoder spans the same subspace as principal component analysis (PCA). The latent codes coincide with PCA scores up to an orthogonal transform [50]. Thus, AEs generalize PCA from linear to nonlinear manifolds. Since PCA identifies the directions of maximum variance and yields a linear approximation of the data, it is mathematically equivalent with POD, which applies the same SVD principle to dynamical system snapshots. AEs can therefore be viewed as a nonlinear generalization of classical projection-based reduction methods, extending their applicability to problems with strongly nonlinear dynamics or parameter dependencies.

For high-dimensional spatial fields, convolutional autoencoders (CAEs) replace dense layers with convolutions in the encoder and transpose-convolutions or upsampling in the decoder [51]. As in any CNN, local connectivity and weight sharing yield far fewer parameters and exploit spatial correlations, which is not only advantageous for images, but can also be used for flow fields in fluid dynamics simulations [47], [52].

2.2.2 POD-enhanced deep-learning ROMs

While POD provides a systematic way to reduce the dimensionality of spatial fields, many modes may be required to capture complex dynamics with sufficient accuracy. Deep learning methods such as autoencoders can alleviate this issue by learning nonlinear trial manifolds. However, training a network to map input parameters directly to high-dimensional CFD fields is computationally prohibitive, as both the network size and training cost scale with the output dimension. POD-DL-ROMs combine the strengths of both approaches: POD is used as a first compression step, after which deep neural networks learn a compact nonlinear representation and the associated dynamics [18].

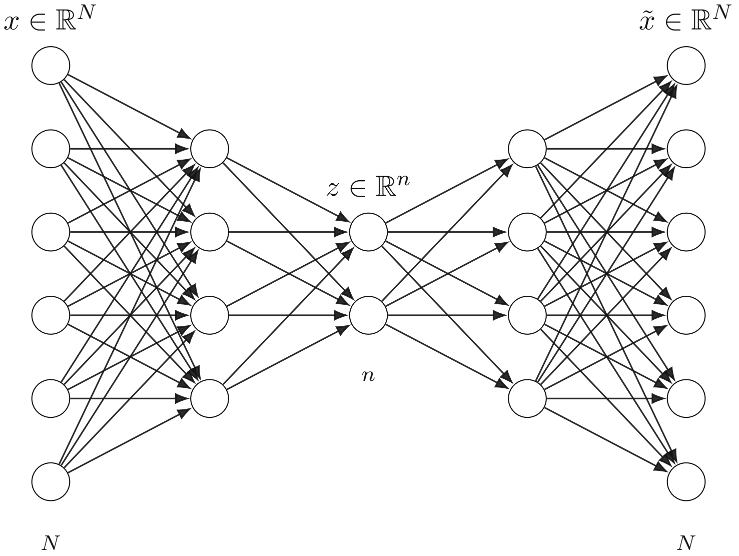


Fig. 2.3: Autoencoder architecture. The encoder maps the high-dimensional input $x \in \mathbb{R}^N$ to a low-dimensional latent code $z \in \mathbb{R}^n$ with $n \ll N$; the decoder reconstructs the approximation $\tilde{x}$ from z.

Given high-fidelity CFD snapshots $u_h(t;\mu) \in \mathbb{R}^{N_h}$, where parameters $\mu \in \mathbb{R}^{n_\mu}$ and $t \in [0,T)$, we first apply (2.1) to construct a POD basis $V_N \in \mathbb{R}^{N_h \times N}$, $N \ll N_h$ and compute the POD coefficients $u_N(t;\mu) \in \mathbb{R}^N$. After this first linear dimensionality reduction, we further reduce the dimension by training an autoencoder consisting of an encoder E_θ and decoder D_ϕ. In parallel, a multi-layer perceptron $F_\psi : \mathbb{R}^{n_\mu+1} \to \mathbb{R}^n$ is trained to map parameters and time directly to the latent coordinates:

$$z(t;\mu) \approx F_\psi(t,\mu). \tag{2.9}$$

The networks are trained jointly by minimizing a loss function that couples latent-space alignment with POD reconstruction:

$$\mathcal{L}(\theta,\phi,\psi) = \frac{1}{N_s}\sum_{j=1}^{N_s}\Big(\lambda_z \left\|E_\theta(u_N^{(j)}) - F_\psi(t^{(j)},\mu^{(j)})\right\|_2^2 + \lambda_u \left\|u_N^{(j)} - D_\phi(F_\psi(t^{(j)},\mu^{(j)}))\right\|_2^2\Big), \tag{2.10}$$

where $\lambda_z, \lambda_u > 0$ are weighting factors. The first term enforces consistency between the autoencoder encoder and the dynamics network in latent space, while the second term enforces accurate reconstruction of the POD coefficients from predicted latent variables.

At inference time, the POD–DL–ROM surrogate reconstructs the high-fidelity solution by concatenating the networks with the POD basis:

$$\hat{u}_h(t;\mu) = V_N\, D_\phi\big(F_\psi(t,\mu)\big). \tag{2.11}$$

The resulting models predict high-dimensional CFD fields with high accuracy while remaining efficient to train and evaluate in terms of computational cost and memory. This makes them particularly well suited for ensemble-based uncertainty quantification, as discussed in the next section.

2.3 Uncertainty Quantification

In reduced-order modeling it is important to report both accurate predictions and the uncertainty of those predictions. Predictive uncertainty has two sources: epistemic uncertainty (limited data and modeling choices) and aleatoric uncertainty (noise and variability in the data) [23].

Bayesian view of epistemic uncertainty Specifically in neural networks, the epistemic uncertainty refers to the uncertainty due to a model's parameters. For quantifying it, a Bayesian perspective is commonly adopted [53]. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training set and $w \in \mathbb{R}^{d_w}$ denote the network parameters. In a Bayesian formulation, the parameters w are random variables with posterior

$$p(w \mid \mathcal{D}) \;\propto\; p(\mathcal{D} \mid w)\, p(w), \tag{2.12}$$

where $p(w)$ is a prior and $p(\mathcal{D} \mid w)$ is the likelihood. For a new input x^*, the predictive distribution for the epistemic uncertainty marginalizes the weights,

$$p(y^* \mid x^*, \mathcal{D}) \;=\; \int p(y^* \mid x^*, w)\; p(w \mid \mathcal{D})\, dw. \tag{2.13}$$

For deep neural networks, the posterior in (2.12) is generally intractable, so we use approximations.

Aleatoric uncertainty via heteroscedastic likelihoods Depending on the data, aleatoric uncertainty can be assumed homoscedastic, i.e., constant for every input x, or heteroscedastic. The latter assumes that the uncertainty depends on the input [23]. For regression problems, neural networks typically predict a single value and parameters are learned by minimizing the mean squared error (MSE) on the training set, i.e., $\sum_{n=1}^{N} (y_n - \hat{y}(x_n))^2$. However, this does not capture aleatoric uncertainty. To this end, we assume that the data is generated by a Gaussian distribution and change our network to predict the mean $\mu(x)$ and the variance $\sigma^2(x)$ in the last layer. In such a model, the loss is replaced by the negative log-likelihood loss function, which can be seen as a heteroscedastic generalization of the MSE

$$\mathcal{L}_{NN}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{||y_i - \mu_\theta(x_i)||^2}{2\sigma_\theta^2(x_i)} + \frac{1}{2} \log \sigma_\theta^2(x_i) \right) \tag{2.14}$$

for a diagonal $\Sigma_\theta(x) = \mathrm{diag}(\sigma_{\theta,1}^2(x), \ldots)$. This approach equates to performing a maximum likelihood (MLE) inference and finds a single value for the model parameters θ and therefore does not account for epistemic uncertainty.

Deep ensembles for total predictive uncertainty To combine both types of uncertainty, we train an ensemble of heteroscedastic models, a so-called deep ensemble [21]. Deep ensembles can be interpreted as an empirical approximation of the weight posterior and, consequently, as a way to estimate epistemic uncertainty. Training M independent heteroscedastic models to obtain $(\mu_m(x), \Sigma_m(x))_{m=1}^{M}$ as an approximation of the posterior predictive distribution in (2.13)

$$p(y \mid x, \mathcal{D}) \approx \frac{1}{M} \sum_{m=1}^{M} \mathcal{N}(y; \mu_m(x), \Sigma_m(x)). \tag{2.15}$$

Using this equation, the mean of the predictive distribution can be computed using the law of total expectation as

$$\mathbb{E}[y \mid x, \mathcal{D}] = \frac{1}{M} \sum_{m=1}^{M} \mu_m(x) \tag{2.16}$$

and the covariance is given by the law of total variance

$$\operatorname{Cov}[y \mid x, \mathcal{D}] = \underbrace{\frac{1}{M} \sum_{m=1}^{M} \Sigma_m(x)}_{\text{aleatoric}} + \underbrace{\frac{1}{M} \sum_{m=1}^{M} \big(\mu_m(x) - \mathbb{E}(x)\big)\big(\mu_m(x) - \mathbb{E}(x)\big)^{\top}}_{\text{epistemic}} . \tag{2.17}$$

2.4 Specialization to the POD-DL-ROM

To enable the POD–DL–ROM to predict aleatoric uncertainty while maintaining its ability to learn latent dynamics as before, we extend the decoder network D_ϕ with a heteroscedastic output. Instead of predicting only the mean reconstruction $\mu_\phi(z)$, the decoder now produces both the mean and a corresponding variance $\sigma_\phi^2(z)$ for each POD mode, where the variance represents data-dependent noise in the reduced coefficients. The encoder E_θ and the reduced dynamics network F_ψ remain unchanged, with F_ψ still mapping temporal and parametric inputs (t, μ) to the latent representation z.

During training, the deterministic reconstruction loss in the original POD–DL–ROM formulation (eq. 2.10) is replaced by a negative log-likelihood (NLL) loss that penalizes both reconstruction error and predictive uncertainty. The resulting combined loss function reads

$$\mathcal{L}(\theta, \phi, \psi) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left[\lambda_z \left\| E_\theta(u_N^{(j)}) - F_\psi(t^{(j)}, \mu^{(j)}) \right\|_2^2 + \lambda_u \left(\frac{\left\| u_N^{(j)} - \mu_\phi(z^{(j)}) \right\|^2}{2\, \sigma_\phi^2(z^{(j)})} + \frac{1}{2} \log \sigma_\phi^2(z^{(j)}) \right) \right] , \tag{2.18}$$

where $z^{(j)} = F_\psi(t^{(j)}, \mu^{(j)})$ and D_ϕ outputs $\mu_\phi(\cdot)$ and $\sigma_\phi^2(\cdot)$ as the mean and variance of the predicted POD coefficients. The NLL term encourages the model to express higher uncertainty in regions with larger reconstruction errors, allowing the network to distinguish between confidently and uncertainly predicted flow states.

The network predicts the reduced coefficients $u_N(t; \mu) \in \mathbb{R}^N$ as a random vector parameterized by the decoder outputs. For an ensemble of models $\{(\mu_m, \Sigma_m)\}_{m=1}^{M}$, equations (2.16)–(2.17) yield the aggregated predictive mean and covariance, $\mathbb{E}[u_N \mid x, \mathcal{D}]$ and $\operatorname{Cov}[u_N \mid x, \mathcal{D}]$, respectively. Projecting these quantities back to the high-dimensional space using the POD basis $V_N \in \mathbb{R}^{N_h \times N}$ gives

$$\mathbb{E}[u_h \mid x, \mathcal{D}] = V_N\, \mathbb{E}[u_N \mid x, \mathcal{D}], \qquad \operatorname{Cov}[u_h \mid x, \mathcal{D}] = V_N \operatorname{Cov}[u_N \mid x, \mathcal{D}]\, V_N^{\top} . \tag{2.19}$$

Even if each ensemble member predicts a diagonal covariance Σ_m in POD space, eq. (2.19) induces a spatial covariance structure in physical space through the linear mapping by V_N. The combination of the POD basis with heteroscedastic decoder heads and deep ensembles therefore produces spatially correlated uncertainty fields.

Chapter 3

Results

We evaluated the proposed POD-DL-ROM with uncertainty quantification on the same benchmark families as [18] while providing an assessment of the predictive uncertainty. Concretely, we compare results for three parameterized CFD problems and against the POD-DL-ROM accuracy baselines in the literature. Our evaluation protocol follows [18]: for each test parameter $\mu_{\text{test},i}$ we reconstructed the full time trajectory and computed the error indicator $\epsilon_{rel} \in \mathbb{R}$ aggregated over time and over the test set:

$$\epsilon_{rel} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(\frac{\sqrt{\sum_{k=1}^{N_t} \|u_h^k(\mu_{test,i}) - \tilde{u}_h^k(\mu_{test,i})\|^2}}{\sqrt{\sum_{k=1}^{N_t} \|u_h^k(\mu_{test,i})\|^2}} \right) \tag{3.1}$$

and the maximum error $\epsilon_{max} \in \mathbb{R}$:

$$\epsilon_{max} = \max_{i=1,\ldots,N_{test}} \left(\frac{\sqrt{\sum_{k=1}^{N_t} \|u_h^k(\mu_{test,i}) - \tilde{u}_h^k(\mu_{test,i})\|^2}}{\sqrt{\sum_{k=1}^{N_t} \|u_h^k(\mu_{test,i})\|^2}} \right). \tag{3.2}$$

For individual cases, we additionally define the relative error e_k

$$\epsilon_k = \frac{[u_h^k(\mu_{test)} - \tilde{u}_h^k(\mu_{test})]}{\sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} \|u_h^k(\mu_{test})\|^2}}. \tag{3.3}$$

Besides these error metrics comparing the accuracy of the mean predictions, we also assessed the quality of the uncertainty estimates. In recent years, several metrics have been proposed to evaluate the quality of uncertainty quantification in regression tasks. Based on the discussion in [54], we selected the Area Under the Sparsification Error Curve and the Calibration Error for regression tasks.

Area Under the Sparsification Error Curve (AUSE) The AUSE metric is based on the sparsification error (SE) curve, which is derived from the sparsification plot. The sparsification plot generally consists of two curves: One is most commonly referred to as the *oracle* and reflects the overall model error when progressively removing the data point with the highest residual. The second curve also represents the model error, but it is generated by progressively removing data points with the highest predicted uncertainty instead of the error. The sparsification plot enables an easy comparison of these two curves and therefore an easy visual assessment of whether higher predicted uncertainty corresponds with higher error values. The AUSE metric, as proposed by Ilg et al. [55] and formalized by Kristoffersson Lind et al. [54], is then computed as the area between the sparsification curve created from the model's uncertainty and the oracle sparsification curve. A lower AUSE value indicates better uncertainty estimation, as it implies that uncertainty predictions are closely aligned with the actual errors.

Formally, we define the per-sample squared field error

$$e_i^k = \left\|u_h^k(\mu_{\text{test},i}) - \tilde{u}_h^k(\mu_{\text{test},i})\right\|^2, \tag{3.4}$$

and the per-sample uncertainty score as total predictive variance

$$s_i^k = \operatorname{tr}(\tilde{\Sigma}_i^k) \tag{3.5}$$

for any given $k \in \{1, \dots, N_t\}$. Let $S_{test} = \{(\mu_{\text{test},i},\ u_h^k(\mu_{\text{test},i})) : i = 1, \dots, N_{\text{test}},\ k = 1, \dots, N_t\}$ denote the evaluation set. We measure prediction error with

$$\operatorname{MSE}(S) := \frac{1}{|S|} \sum_{i \in S} e_i^k. \tag{3.6}$$

For a retention fraction $\alpha \in [0, 1]$, we construct the subset $S_{err}(\alpha)$ of S_{test} by picking the samples with the smallest error e_i^k such that $|S_{AE}(\alpha)| = \alpha N_{test}$. Analogously, we define $S_U(\alpha)$ by picking samples with smallest predicted uncertainty s_i^k. The Area Under the Sparsification Error curve (AUSE) is then given by

$$\operatorname{AUSE}(S_{test}) = \int_0^1 \left[\frac{\operatorname{MSE}(S_U(\alpha))}{\operatorname{MSE}(S_{test})} - \frac{\operatorname{MSE}(S_{err}(\alpha))}{\operatorname{MSE}(S_{test})}\right] d\alpha. \tag{3.7}$$

The integral measures the normalized area between the sparsification curve based on predicted uncertainties and the oracle curve based on true errors. In practice, the integral in Eq. (3.7) is approximated by a discrete sum over uniformly spaced values of $\alpha \in [0, 1]$. The AUSE is dimensionless, and a value of zero corresponds to perfect alignment between predicted and true uncertainty orderings.

Calibration Error (CE) In addition to the AUSE, we also report the calibration error. It is derived from the calibration plot [56], which compares the predicted probabilities that the true values fall below certain quantiles with their empirical frequencies. A lower calibration error indicates that the predicted uncertainty distributions are statistically consistent with the observed errors, i.e., the model's predicted variances are well calibrated. Originally proposed for classification tasks [57], we adapt the calibration error to regression following [58].

For each test parameter $\mu_{\text{test},i}$ and time step t_k, the model predicts a Gaussian distribution $\mathcal{N}(\hat{u}_h^k(\mu_{\text{test},i}),\ \hat{\sigma}_i^{k\,2})$. Let $F_i^{(k)} : \mathbb{R} \to [0, 1]$ denote its cumulative distribution function

$$F_i^{(k)}(y) = \Phi\left(\frac{y - \hat{u}_h^k(\mu_{\text{test},i})}{\hat{\sigma}_i^k}\right) \tag{3.8}$$

and $F_i^{(k)\,-1}(p)$ its inverse (quantile) function.

A perfectly calibrated model satisfies that the empirical frequency with which the true field values fall below the predicted quantile $F^{-1}(p)$ equals p:

$$\mathbb{E}\Big[\mathbf{1}\Big\{u_h^k(\mu_{\text{test},i}) \le F_i^{(k)\,-1}(p)\Big\}\Big] = p, \qquad p \in [0, 1]. \tag{3.9}$$

We empirically estimate this relation for all test parameters and time steps. The empirical coverage curve is

$$\hat{c}(p) := \frac{1}{N_{\text{test}}\, N_t} \sum_{i=1}^{N_{\text{test}}} \sum_{k=1}^{N_t} \mathbf{1}\Big\{u_h^k(\mu_{\text{test},i}) \le F_i^{(k)\,-1}(p)\Big\}. \tag{3.10}$$

Perfect calibration corresponds to $\hat{c}(p) = p$ for all p. The Calibration Error (CE) quantifies the deviation from this ideal:

$$\mathrm{CE} = \int_0^1 \left|\hat{c}(p) - p\right| dp \;\approx\; \sum_{m=1}^{K} \left|\hat{c}(p_m) - p_m\right| \Delta p, \tag{3.11}$$

where $\{p_m\}_{m=1}^K \subset (0, 1)$ is a uniform grid with $\Delta p = 1/K$. CE is dimensionless and equals 0 for a perfectly calibrated model. Larger values indicate miscalibration, that is, over- or under-estimation of predictive uncertainty.

3.1 Common setup

3.1.1 FOM snapshots

All FOM simulations were performed using the finite–volume method (FVM) implemented in OpenFOAM [59]. Specifically, we used the open–source community version 12. The transient, incompressible Navier–Stokes equations were solved using the `incompressibleFluid` solver. Pressure equations were solved with the `GAMG` iterative solver and `DIC` preconditioning, while velocity and scalar fields were solved with the `PBiCGStab` solver and `DILU` preconditioning. All fields allowed for a residual of 10^{-6}.

The geometries were meshed using the utilities provided by OpenFOAM. Mesh quality (non-orthogonality $< 70°$, skewness < 4) was verified using OpenFOAM's internal mesh checks.

All FOM simulations were executed on a MacBook Pro with an M1 Pro chip and 32 GB of memory.

3.1.2 ROM training

The POD-DL-ROM is implemented using PyTorch [60]. We set the weighting factors in equation 2.10 to $\lambda_z, \lambda_u = 0.5$ for all the cases presented, as suggested by [18]. The POD dimension N is set to 64, balancing accuracy and computational efficiency. The resulting POD-reduced snapshots were scaled using Min-Max scaling to the range $[0, 1]$. In contrast to [18], we applied this scaling to each POD mode individually, rather than applying the same scaling to all modes of a given channel. In our experiments, this approach has shown to improve the accuracy of the results. The training of the model was performed using the Adam optimizer with a learning rate of 0.001. For the details of the training algorithm, we refer to [18]. For every case, an ensemble of eight POD-DL-ROMs was trained for up to 2500 epochs each, with early-stopping if no improvement in validation loss took place over the course of 500 epochs. Compared to [18], the lower maximum number of epochs was selected to partly offset the increased computational cost from training an ensemble. The batch size was set to 128. Training was performed on NVIDIA A100 GPUs within the Dutch HPC cluster *Snellius*.

3.2 Unsteady advection-diffusion-reaction equation

As a first benchmark, we consider the unsteady advection–diffusion–reaction system with the solution $u = u(t, x; \mu)$ defined on the spatial domain $\Omega = [0, 1]^2$ and the time interval $[0, T]$ with $T = 10\pi$. The governing equation is

$$\frac{\partial u}{\partial t} - \nabla \cdot (\mu_1 \nabla u) + b(t; \mu_2) \cdot \nabla u + cu = f(\mu_3, \mu_4), \quad (t, x) \in [0, T] \times \Omega, \tag{3.12}$$

subject to the initial condition $u(0, x; \mu) = 0$ and homogeneous Neumann boundary conditions $\mu_1 \nabla u \cdot n = 0$ on $\partial\Omega$. The source term is defined as

$$f(x, y; \mu_3, \mu_4) = 10 \exp\left(-\frac{(x - \mu_3)^2 + (y - \mu_4)^2}{0.07^2} \right), \tag{3.13}$$

and the time-dependent advection field reads

$$b(t; \mu_2) = \begin{bmatrix} \cos\left(\frac{\pi}{\mu_2 t}\right) \\ \sin\left(\frac{\pi}{\mu_2 t}\right) \end{bmatrix}. \tag{3.14}$$

The parameter vector $\mu = (\mu_1, \mu_2, \mu_3, \mu_4) \in \mathbb{R}^4$ comprises the diffusion coefficient $\mu_1 \in [0.002, 0.005]$, the amplitude of the advection field $\mu_2 \in [30, 70]$, and the source coordinates $\mu_3, \mu_4 \in [0.4, 0.6]$. The full-order model (FOM) is discretized using the finite volume method on a uniform grid with $N_h = 10.000$ cells and a time step $\Delta t = 2\pi/20$, leading to $N_t = 100$ time instances.

To generate training data, we sample $N_{\text{train}} = 500$ parameter instances uniformly from the parameter space. The test set contains $N_{\text{test}} = 192$ parameter instances drawn from the same distribution. The latent dimension is set to $n = n_\mu + 1 = 5$, corresponding to one additional degree of freedom beyond the number of input parameters.

After training for 3 hours and 6 minutes, the ROM ensemble predicted the corresponding temporal evolution in 0.10 s per case while a full FOM simulation for one parameter instance required 27.5 s, yielding a 286× speedup (Table 3.2).

Accuracy The ensemble achieves a mean relative error of $e_{\text{rel}} = 4.3 \times 10^{-2}$ and a maximum error of $e_{\text{max}} = 1.13 \times 10^{-1}$ on the test set. Figure 3.1 illustrates the comparison between the FOM and the predicted ROM solution for two representative parameter instances at $t = 28.9$ and $t = 27.6$. The ROM accurately predicts the overall shape and amplitude of the transported scalar field, demonstrating that the model captures both the diffusive and advective effects of the system well.

Uncertainty The bottom row of Figure 3.1 shows the overall predicted uncertainty. Regions of high uncertainty coincide with regions of high relative error, indicating that the model correctly identifies areas where its predictions are less reliable. The average uncertainty sparsification error is $AUSE = 0.091$ and the calibration error is $CE = 0.077$. These low but non-zero values are confirmed by the diagnostic plots in Figure 3.2: in the calibration plot (left), the empirical coverage curve lies below the diagonal for $p \in [0.1, 0.5]$ and above it for $p \in [0.5, 1]$, indicating slight underconfidence for narrow and slight overconfidence for wide predictive intervals. The sparsification curve (center) decreases as uncertain samples are removed, showing that the predicted variances are informative. The sparsification error (right) remains low for most retention fractions, suggesting a consistent ordering of uncertainty magnitudes across the field.

3.3 Von Kármán vortex street

As a second benchmark, we consider the unsteady incompressible Navier-Stokes equations in primitive variables, i.e., velocity $\mathbf{v}$ and pressure p. The test case describes the two-dimensional laminar flow past a circular cylinder, a standard benchmark for the evaluation of numerical algorithms for incompressible flow [18]. The governing equations read

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} - \nabla\cdot\boldsymbol{\sigma}(\mathbf{v},p) = \mathbf{0}, \quad (x,t)\in\Omega^F\times(0,T), \tag{3.15}$$

$$\nabla\cdot\mathbf{v} = 0, \quad (x,t)\in\Omega^F\times(0,T), \tag{3.16}$$

with boundary and initial conditions

$$\mathbf{v} = \mathbf{0} \qquad \text{on } \Gamma_{D1}\times(0,T), \tag{3.17}$$

$$\mathbf{v} = \mathbf{h}(x,t;\mu) \qquad \text{on } \Gamma_{D2}\times(0,T), \tag{3.18}$$

$$\boldsymbol{\sigma}(\mathbf{v},p)\mathbf{n} = \mathbf{0} \qquad \text{on } \Gamma_N\times(0,T), \tag{3.19}$$

$$\mathbf{v}(\cdot,0) = \mathbf{0} \qquad \text{in } \Omega^F. \tag{3.20}$$

Here ρ denotes the fluid density and $\boldsymbol{\sigma}$ the Cauchy stress tensor,

$$\boldsymbol{\sigma}(\mathbf{v},p) = -p\mathbf{I} + 2\nu\,\mathbf{e}(\mathbf{v}), \qquad \mathbf{e}(\mathbf{v}) = \tfrac{1}{2}(\nabla\mathbf{v} + (\nabla\mathbf{v})^\top). \tag{3.21}$$

The inflow profile is prescribed as

$$\mathbf{h}(x,t;\mu) = \begin{bmatrix} \dfrac{4\,U(t;\mu)\,x_2(0.41-x_2)}{0.41^2} \\ 0 \end{bmatrix}, \qquad U(t;\mu) = \mu\,\sin\Big(\frac{\pi t}{8}\Big), \tag{3.22}$$

so that the velocity oscillates sinusoidally in time.

The computational domain represents a two-dimensional channel with a circular obstacle, $\Omega^F = (0,2.2)\times(0,0.41)\setminus\overline{B}_{0.05}(0.2,0.2)$, where $B_r(\mathbf{x}_c)$ denotes a disk of radius r centered at $\mathbf{x}_c$. The boundary is decomposed into $\partial\Omega^F = \Gamma_{D1}\cup\Gamma_{D2}\cup\Gamma_N$, where $\Gamma_{D1} = \{x_2 = 0\}\cup\{x_2 = 0.41\}\cup\partial B_{0.05}(0.2,0.2)$, $\Gamma_{D2} = \{x_1 = 0, x_2\in[0,0.41]\}$, and $\Gamma_N = \{x_1 = 2.2, x_2\in[0,0.41]\}$. Here, $\mathbf{n}$ denotes the outward unit normal vector on $\partial\Omega^F$.

We set the fluid density to $\rho = 1\,\mathrm{kg/m^3}$ and the kinematic viscosity to $\nu = 10^{-3}\,\mathrm{m^2/s}$. The parameter $\mu\in[1,2]\,\mathrm{m/s}$ controls the maximum inflow velocity, corresponding to Reynolds numbers $\mathrm{Re}\in[66,133]$. We solve the FOM on a mesh with $N_h = 8.224$ cells. Time integration uses a second–order implicit backward scheme (BDF2) with fixed step $\Delta t = 2\times10^{-3}\,\mathrm{s}$ up to $T = 8\,\mathrm{s}$. The convective term in the momentum equation is discretized by Gauss–linear differencing, and diffusive operators use an orthogonality–corrected Laplacian; gradients use least–squares. This setup yields the expected laminar von Kármán vortex shedding downstream of the cylinder at appropriate Reynolds numbers.

In contrast to the scalar-transport case, the model here is set up so that it learns to predict a three-channel field (two velocity and a single pressure component). We select again $n = 5$ as the latent dimension of the autoencoder in accordance with [61]. We use $N_{train} = 21$ training parameter instances and $N_{test} = 20$ testing parameter instances uniformly distributed over the parameter range. For each parameter set, $N_t = 400$ time instances are sampled.

After an average training time of 49.8 minutes per ensemble member, the ensemble reproduced each parameter instance in 1.32 s compared to 111.3 s for the FOM, corresponding to an 84× reduction in wall-clock time (Table 3.2).

Accuracy Despite the more complex flow dynamics, the ensemble reproduces the flow with low error. Aggregated over the test trajectories, the relative errors for velocity magnitude and pressure are $e_{\mathrm{rel}} = 3.70\times10^{-2}$ and $e_{\mathrm{rel}} = 4.59\times10^{-2}$, while the maximal trajectory errors are $e_{\max} = 8.3 \times 10^{-2}$ and $e_{\max} = 1.4 \times 10^{-1}$, respectively. Figures 3.4 & 3.5 illustrate two exemplary cases. For a low inflow velocity without pronounced shedding (Fig. 3.4), both $|\mathbf{v}|$ and p are matched well across the domain. The residuals of the velocity field are generally low and distributed across the domain, with the extrema appearing around the cylinder boundary and the start of the shear layers. The residuals of the pressure field correlate with the development of the pressure across the case, the highest residuals are at the inlet and around the obstacle and reduce to a minimum at the outlet.

In a case with pronounced shedding (Fig. 3.5), the ROM correctly predicts the vortices of the velocity field and the associated pressure hotspots. The residuals are generally higher compared to the previous case and are the highest in regions where the vortex shedding takes place, becoming smaller further down the channel. It is interesting that in this case, the residuals around the obstacle are comparatively small compared to the residuals in the region of vortex shedding.

Uncertainty The lowest row in figure 3.4 and 3.5 provides again insight into the field uncertainty of the model. As in the scalar transport case, the predicted standard deviation correlates with the residuals. For $|\mathbf{v}|$, regions of high uncertainty in the velocity field are mostly concentrated after the obstacle in the area of the first vortex shedding. For p it concentrates near the alternating high/low pressure stripes, but is overall more distributed across the case with a lower maximum uncertainty. However, the predicted uncertainty is not as well-calibrated as in the previous case. The calibration error $\mathrm{CE} = 0.1545$ for $|\mathbf{v}|$ and $\mathrm{CE} = 0.181$ for p indicates overconfidence at nominal coverage $p < 0.5$ and underconfidence beyond $p > 0.5$ (see the calibration plots in Figs. 3.6 and 3.7). Nevertheless, the quality of the ranking of uncertainty remains strong as the sparsification curve closely follows the oracle. The area under sparsification error is therefore small, with $\mathrm{AUSE} = 0.0285$ for $|\mathbf{v}|$ and $\mathrm{AUSE} = 0.0492$ for p.

3.4 Cerebral Aneurysm

In the last test case, we consider the simulation of blood flows in a cerebral (intracranial) aneurysm. Cerebral aneurysms are characterized by a localized structural deterioration of the arterial wall, potentially leading to a disruption of the media of the artery [62]. The most severe complication of cerebral aneurysms is their rupture, which causes an intracranial hemorrhage and often leads to death. The prognosis of cerebral aneurysm rupture is however very difficult, while preventative treatments also carry significant risk. Therefore, the development of methods that enable better estimation of the rupture risk of cerebral aneurysms is highly valuable to support clinical decision-making. In the past, aneurysm size has been used as the main indicator for assessing rupture likelihood. However, both large and small aneurysms can rupture, and therefore, research has turned to other factors to be able to better predict the trajectories of aneurysm. Hemodynamics as a potentially informing factor in the development, progression, and rupture of aneurysms has received more interest in recent years [63], [64]. The wall shear stress (WSS) acts directly on the vascular endothelium and seems to play a key role in the development and rupture of aneurysm. However, it is difficult to measure in-vivo. Therefore, CFD simulations based on 3-dimensional CT data and patient-specific hemodynamic information are a key instrument to compute hemodynamic properties in the aneurysm to assist clinicians in their decision-making [65].

For our model, we consider the aneurysm shown in figure 3.8 supported by the AneuX morphology database [66]. Figure 3.9 shows a preprocessed version that we used for all further simulations: a cutout that removes vessels far from the aneurysm dome, remeshed with a target cell area of $0.05mm^2$. We extended the inlet and the outlets using straight tubes with a length $L = 3D$ to allow proper flow development before it reaches the aneurysm dome.

We model the blood flow in the aneurysm domain $\Omega^F \subset \mathbb{R}^3$, whose boundary consists of the arterial wall Γ_w, the inlet Γ_D, and multiple outlets Γ_N, such that $\partial\Omega^F = \Gamma_w \cup \Gamma_D \cup \Gamma_N$. Assuming incompressible, Newtonian flow with constant density $\rho = 1060\,\mathrm{kg/m^3}$ and viscosity $\mu = 3 \times 10^{-3}\,\mathrm{Pa \cdot s}$, and rigid walls, the velocity field $\mathbf{v}(\mathbf{x}, t)$ and pressure field $p(\mathbf{x}, t)$ satisfy the incompressible Navier-Stokes equations in primitive variables:

$$\begin{aligned}
\rho \frac{\partial \mathbf{v}}{\partial t} + \rho\,(\mathbf{v} \cdot \nabla)\mathbf{v} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{v}, p) &= \mathbf{0} && \text{in } \Omega^F \times (0, T), && (3.23)\\
\nabla \cdot \mathbf{v} &= 0 && \text{in } \Omega^F \times (0, T), && (3.24)\\
\mathbf{v} &= \mathbf{0} && \text{on } \Gamma_w \times (0, T), && (3.25)\\
\mathbf{v} &= \mathbf{v}_{\text{in}} && \text{on } \Gamma_D \times (0, T), && (3.26)\\
\boldsymbol{\sigma}(\mathbf{v}, p)\,\mathbf{n} &= \mathbf{0} && \text{on } \Gamma_N \times (0, T), && (3.27)\\
\mathbf{v}(\mathbf{x}, 0) &= \mathbf{0} && \text{in } \Omega^F, && (3.28)
\end{aligned}$$

where the Cauchy stress tensor $\boldsymbol{\sigma}$ is given as in 3.21. At the arterial wall Γ_w, a no-slip condition is imposed. At the inlet Γ_D, we prescribe a pulsatile parabolic velocity profile aligned with the inlet normal $\mathbf{n}_{\text{in}}$ and centered at $\mathbf{x}_c$ on a circular opening of radius $R = H/2$. Let the radial distance on the inlet plane be $r(\mathbf{x}) = \|(\mathbf{I} - \mathbf{n}_{\text{in}}\mathbf{n}_{\text{in}}^\top)(\mathbf{x} - \mathbf{x}_c)\|$. The prescribed profile is

$$\mathbf{v}_{\text{in}}(\mathbf{x}, t) = \begin{cases} u_c(t)\left(1 - \dfrac{r(\mathbf{x})^2}{R^2}\right)\mathbf{n}_{\text{in}}, & r(\mathbf{x}) \le R, \\ \mathbf{0}, & r(\mathbf{x}) > R, \end{cases} \quad \text{on } \Gamma_D \times (0, T), \tag{3.29}$$

with a cosine waveform for the centerline speed

$$u_c(t) = \tfrac{1}{2}\,U_c\,[1 - \cos(2\pi t/T)]. \tag{3.30}$$

While in clinical measurements, the flow in intracranial artieries usually exhibits the characteristic two-peak waveform due to the systolic and diastolic phases of the heartbeat, we selected a simpler cosine waveform since it provides a smooth, strictly periodic inflow with known closed form, which makes it straightforward to evaluate, differentiate, and sample during data generation. A more realistic two-peak waveform would require either measured patient-specific data or a higher-order fitted function. Figure 3.10 shows the profile of the inlet flow rate for selected centerline velocities. The resulting transient laminar flow problem is representative of physiological conditions in medium-sized cerebral arteries, providing a realistic benchmark to evaluate the predictive and uncertainty quantification capabilities of the proposed POD-DL-ROM framework.

The full–order model is discretized using the finite–volume method on an unstructured mesh with $N_h = 12{,}516$ cells. Time integration uses a second–order implicit backward scheme (BDF2) with a constant time step $\Delta t = 10^{-3}\,\mathrm{s}$ over the interval $(0, T)$, with $T = 0.85\,\mathrm{s}$, corresponding to one cardiac cycle. The convective term in the momentum equation is discretized by a Gauss–linear-upwind scheme based on the velocity gradient, while diffusive terms employ an orthogonality–corrected Laplacian formulation. Gradients are computed using a least–squares approach. The single parameter $\mu \in [0.262, 0.398]$ corresponds to the centerline velocity U_c in the inlet boundary condition $\mathbf{v}_{\text{in}}$ and is linearly sampled across the parameter range for both training and testing.

As in previous cases, we set the latent dimension of the POD-DL-ROM to $n = 5$ and sample $N_{\text{train}} = 6$ and $N_{\text{test}} = 3$ parameter instances, considering $N_t = 850$ time steps for each. Since the case is 3-dimensional, the model is constructed to learn to predict four channels: the three velocity components and the pressure for each cell.

After training with an average training time of 26.52 minutes, the ROM achieved a 221× speedup, reducing per-case runtime from 731.6 s for the FOM to 3.30 s (Table 3.2).

Accuracy Figure 3.11 shows the reconstructed velocity and pressure fields for the test parameter $\mu_{\text{test}} = 0.262\,\text{m/s}$ at $t = 0.378\,\text{s}$. It is clearly visible that the ensemble accurately reproduces the main flow features observed in the FOM. As expected from the hemodynamics of aneurysm flow, the velocity magnitude decreases inside the aneurysm dome compared to the parent vessel, while the pressure reaches its maximum near the curvature at the inflow junction.

Quantitatively, the model achieves a mean relative error $e_{\text{rel}} = 0.34 \times 10^{-2}$ for velocity and $e_{\text{rel}} = 0.45 \times 10^{-2}$ for pressure, with corresponding maximum errors $e_{\max} = 0.45 \times 10^{-2}$ and 0.47×10^{-2}. These are the lowest among the presented cases, indicating that the reduced-order model achieves to learn the flow dynamics particularly well in this configuration. This high accuracy can be attributed to the narrow parameter range, which, since it is based on physiologically plausible ranges for the parameter, leads to similar flows across the entire parameter space.

Uncertainty Figure 3.12 illustrates the predicted standard deviation for velocity magnitude and pressure at $\mu_{\text{test}} = 0.262\,\text{m/s}$ and $t = 0.378\,\text{s}$. The distributions of regions with high uncertainty match regions in the aneurysm with high velocity, such as near the inlet curvature and especially after the aneurysm dome, whereas the pressure uncertainty reaches its maxima at the inflow bifurcation and within the aneurysm dome. Areas where the absolute pressure is low also exhibit low uncertainty. The absolute magnitude of the predicted uncertainty, however, remains small, consistent with the low reconstruction errors reported in the previous section.

Figures 3.13 and 3.14 present the corresponding quantitative uncertainty metrics. For the velocity field, the area under the sparsification error is AUSE = 0.39 and the calibration error is CE = 0.18; for the pressure field, the respective values are AUSE = 0.54 and CE = 0.19. The calibration plots reveal an S-shaped curve, which, as in the previous cases, indicates that the model tends to underestimate uncertainty at low nominal probabilities and overestimate it at high ones. The sparsification plots show that the predicted variance only weakly correlates with the true residuals, leading to large sparsification errors.

Tab. 3.1: Unified summary of accuracy and uncertainty metrics across all cases. Accuracy uses ϵ_{rel} and $\epsilon_{\max}$ from Eqs. (3.1)–(3.2). Uncertainty uses AUSE and CE from Eqs. (3.7) and (3.11). Lower is better.

Case	**Field**	$\boldsymbol{\epsilon_{\text{rel}}}$	$\boldsymbol{\epsilon_{\max}}$	**AUSE**	**CE**
Advection–diffusion–reaction	u	4.30×10^{-2}	1.13×10^{-1}	0.091	0.077
Von Kármán vortex street	$\lvert\mathbf{v}\rvert$	3.70×10^{-2}	8.30×10^{-2}	0.028	0.155
	p	4.59×10^{-2}	1.40×10^{-1}	0.049	0.181
Cerebral aneurysm	$\lvert\mathbf{v}\rvert$	0.34×10^{-2}	0.45×10^{-2}	0.390	0.180
	p	0.45×10^{-2}	0.47×10^{-2}	0.540	0.190

Tab. 3.2: Wall-clock runtimes and speedups. A case denotes one parameter combination μ evaluated over all time steps. Training refers to the average training time of each ensemble in the model, as training was executed in parallel across the eight ensemble members. ROM prediction uses an ensemble of eight models evaluated serially.

Case	**FOM per case**	**ROM training**	**ROM per case**	**Speedup**
Advection–diffusion–reaction	27.47 s	3.06 h	0.09 s	286×
Von Kármán vortex street	111.29 s	0.83 h	1.32 s	84×
Cerebral aneurysm	731.62 s	0.44 h	3.30 s	221×

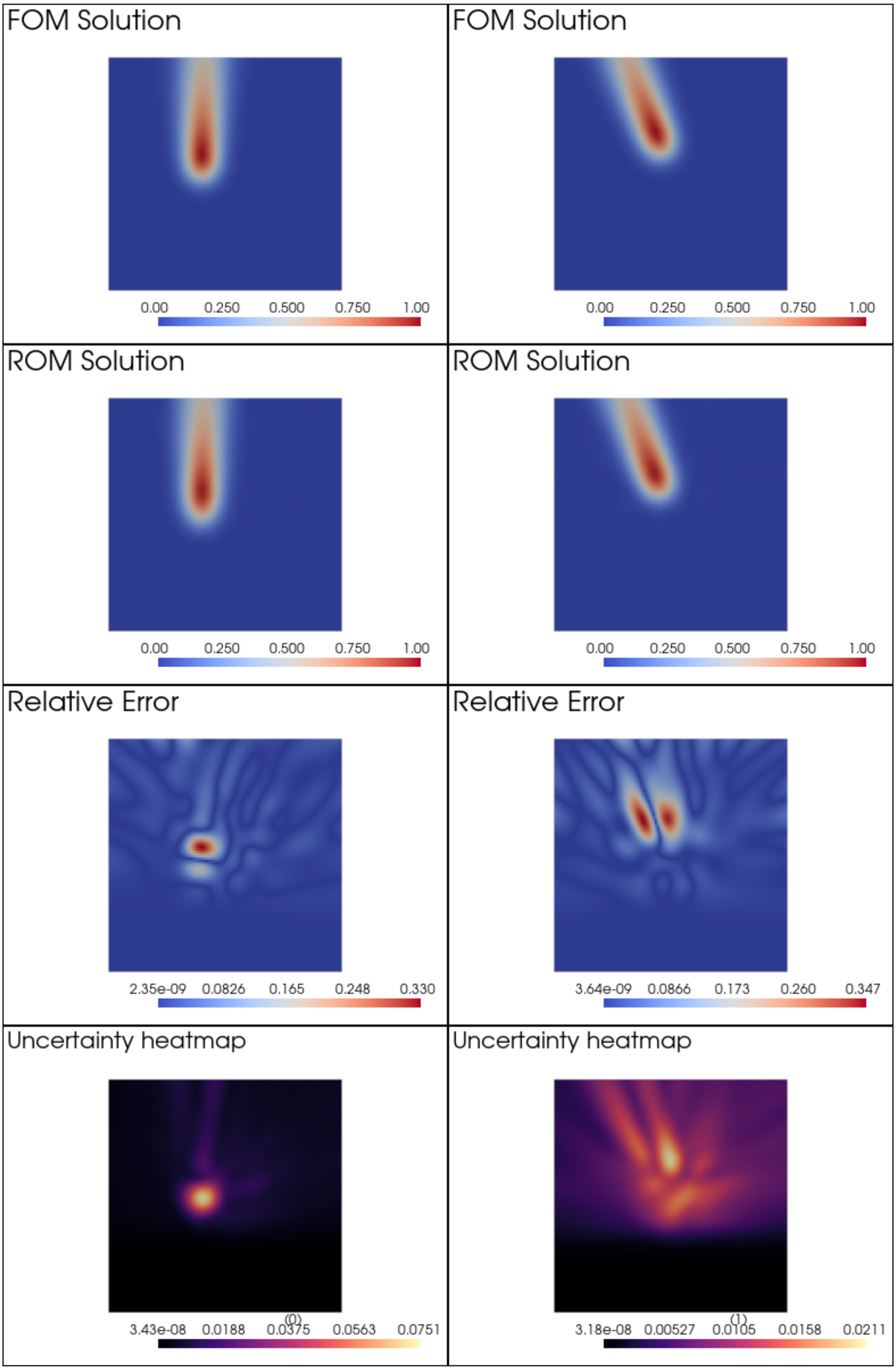


Fig. 3.1: FOM (top row), POD–DL–ROM prediction (second row), relative error (third row), and ensemble variance (bottom row) for test parameter instances $\mu_{\text{test},1} = (0.003, 56.667, 0.4, 0.5)$ at $t = 28.9$ (left) and $\mu_{\text{test},2} = (0.003, 43.333, 0.4667, 0.6)$ at $t = 27.6$ (right).

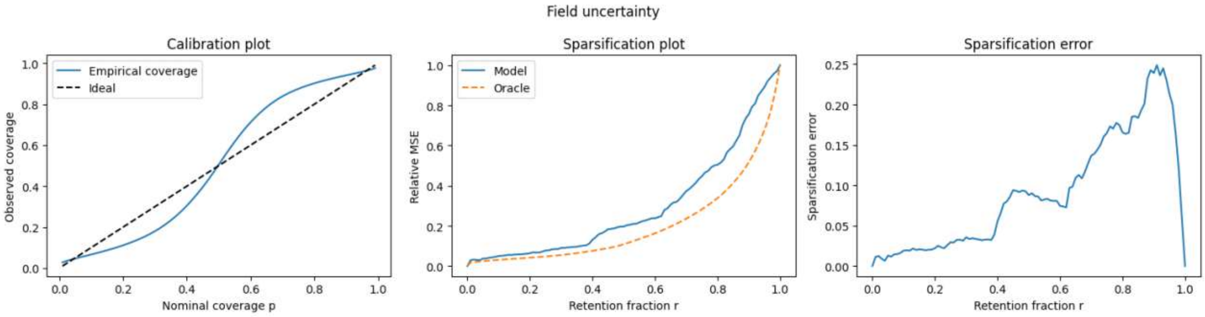


Fig. 3.2: Uncertainty assessment for the advection–diffusion–reaction case: calibration plot (left), sparsification curve (center), and sparsification error (right).

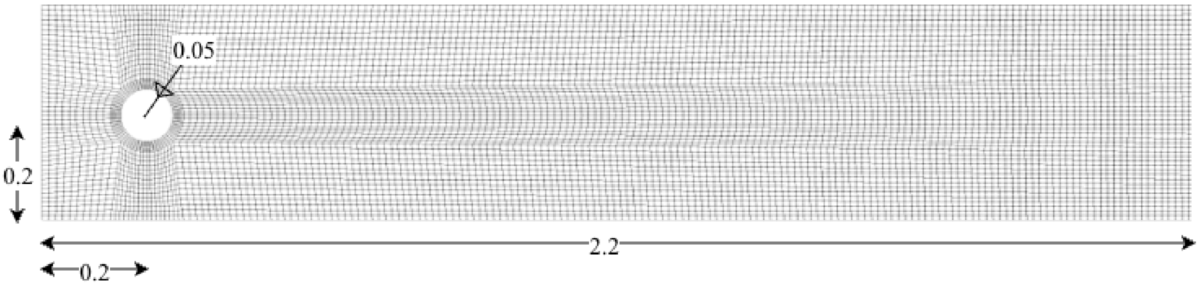


Fig. 3.3: Von Kármán Vortex Street: geometrical configuration and mesh [m]

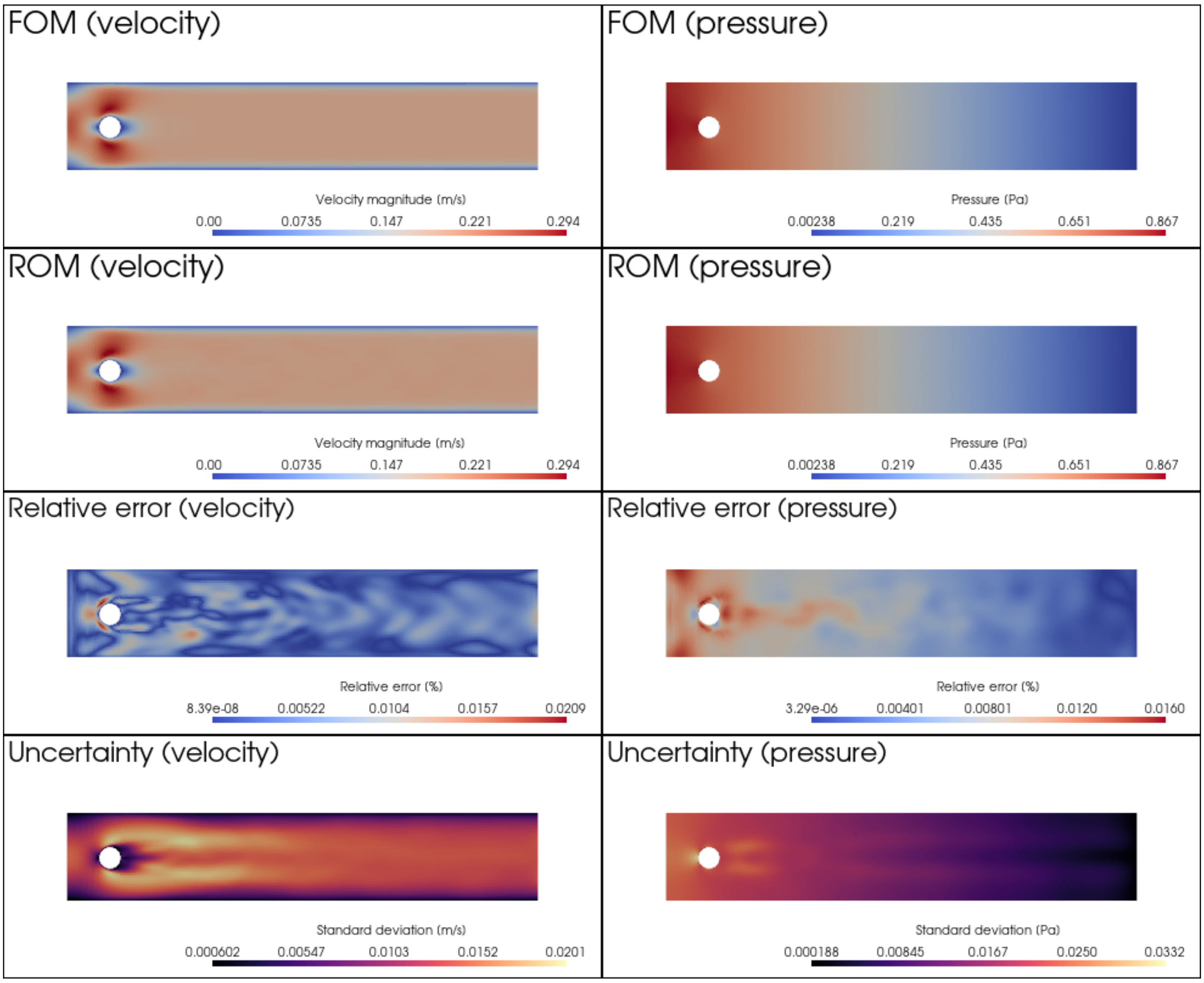


Fig. 3.4: Von Kármán vortex street at $\mu_{\text{test}} = 1.26\,\text{m/s}$ and $t = 0.52\,\text{s}$: FOM (top), POD–DL–ROM (second), relative error (third), and ensemble variance (bottom) for velocity magnitude (left) and pressure (right).

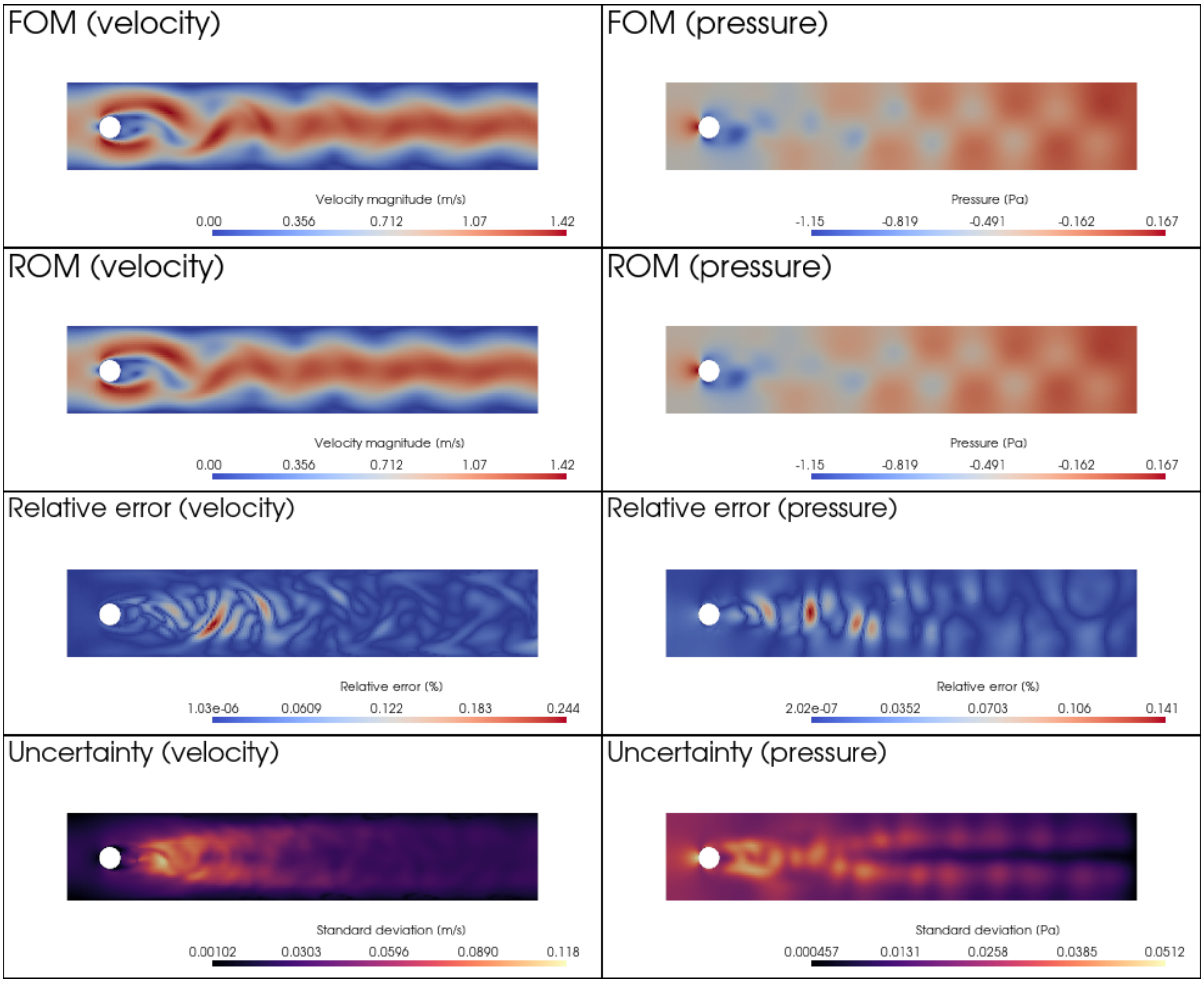


Fig. 3.5: Von Kármán vortex street at $\mu_{\text{test}} = 1.58\,\text{m/s}$ and $t = 6.34\,\text{s}$: same layout as Fig. 3.4.

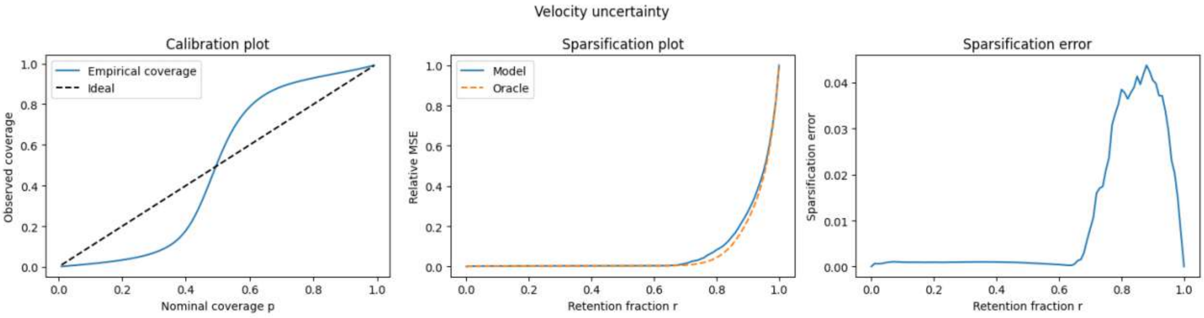


Fig. 3.6: Velocity uncertainty: calibration plot (left), sparsification plot (center), and sparsification error (right). Moderate miscalibration (CE = 0.154) but strong uncertainty ranking (AUSE = 0.0285).

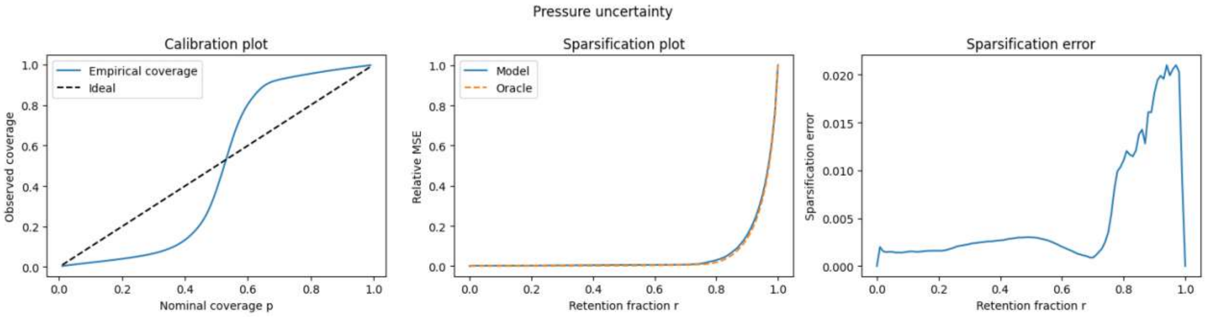


Fig. 3.7: Pressure uncertainty: same metrics as Fig. 3.6. Calibration weaker than velocity (CE = 0.181), yet ranking remains good (AUSE = 0.0492).

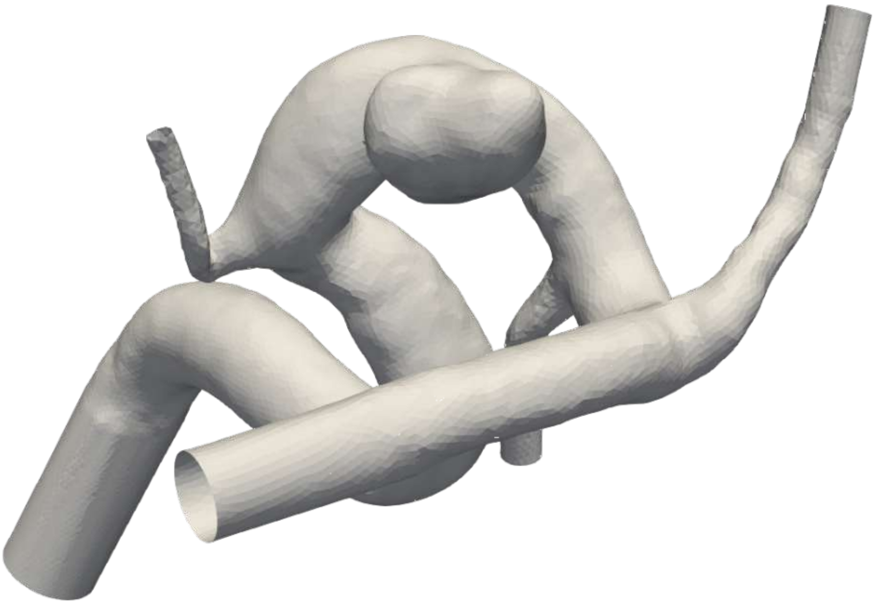


Fig. 3.8: Full cerebral aneurysm geometry with vessels

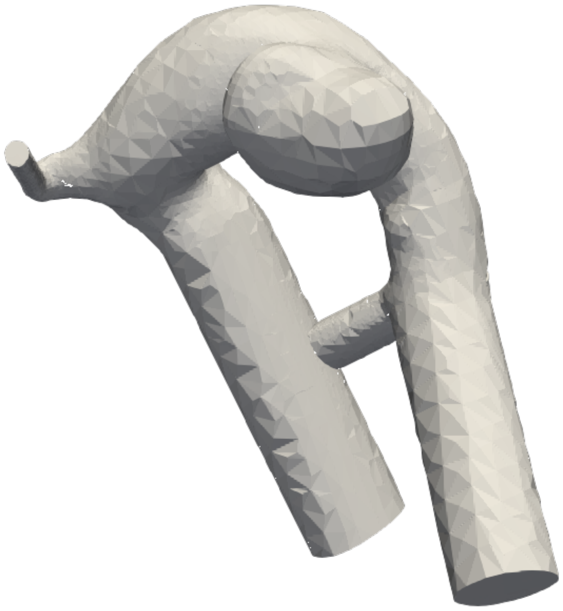


Fig. 3.9: Cerebral aneurysm in configuration *cut2* remeshed to a target cell area of $0.05mm^2$, with inlet and outlets extended by straight sections with $L = 3D$

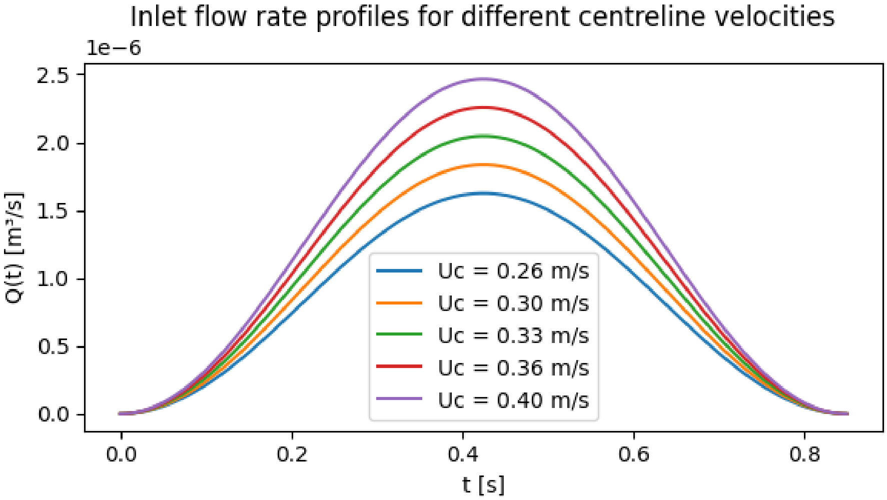


Fig. 3.10: Inlet flow profiles for centreline velocity values $U_c \in \{0.26, 0.30, 0.33, 0.36, 0.4\}m/s$

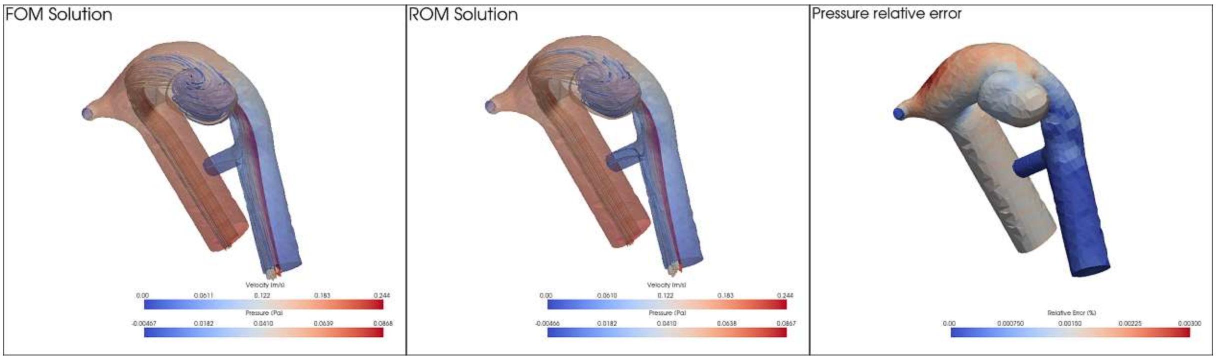


Fig. 3.11: Cerebral aneurysm at $\mu_{\text{test}} = 0.262\,\text{m/s}$ and $t = 0.378\,\text{s}$: FOM (left), POD–DL–ROM (center), and relative pressure error (right). Streamlines represent the velocity field, wall color indicates the pressure.

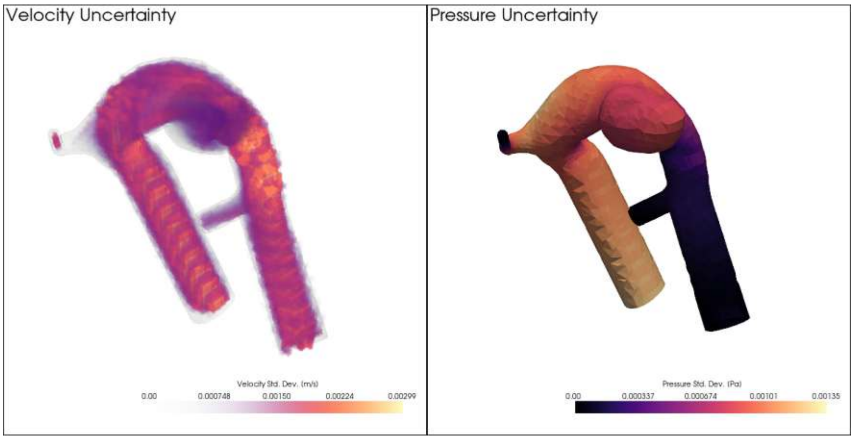


Fig. 3.12: Predicted standard deviation of the velocity magnitude (left) and pressure (right) for the cerebral aneurysm at $\mu_{\text{test}} = 0.262\,\text{m/s}$ and $t = 0.378\,\text{s}$. The velocity uncertainty is rendered volumetrically, while the pressure uncertainty is visualized on the vessel wall.

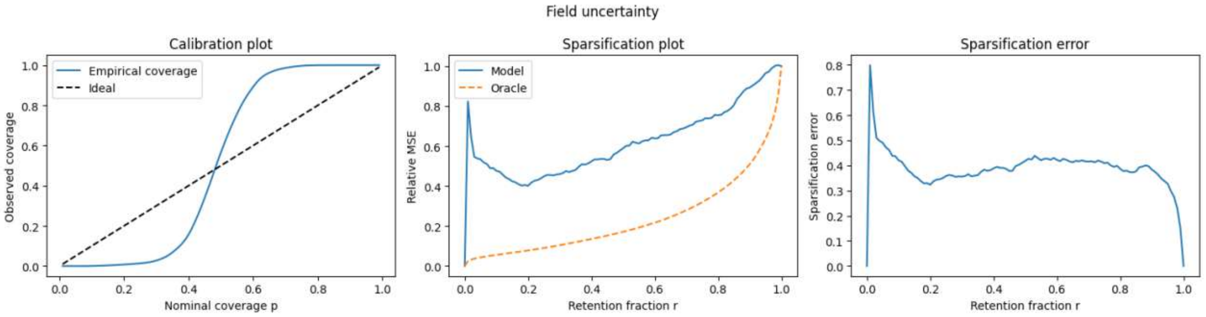


Fig. 3.13: Uncertainty quantification metrics for the velocity field: calibration plot (left), sparsification plot (center), and sparsification error (right).

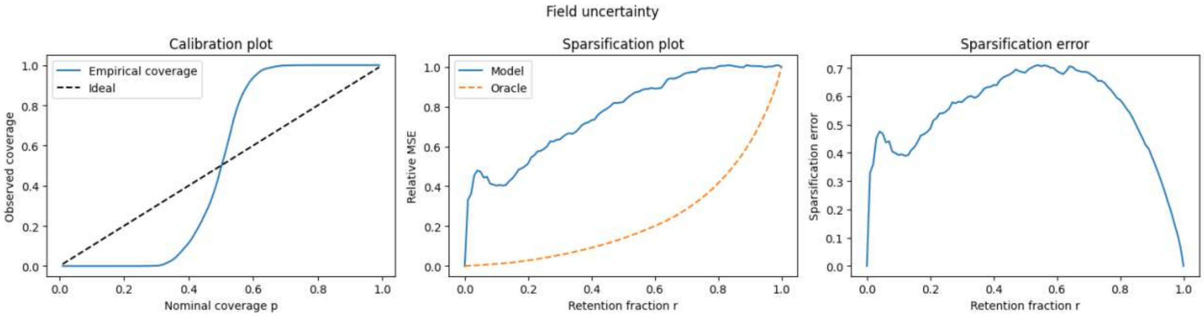


Fig. 3.14: Uncertainty quantification metrics for the pressure field: calibration plot (left), sparsification plot (center), and sparsification error (right).

Chapter 4

Conclusion

In this work, we trained deep ensembles of a modified POD-DL-ROM architecture to assess not only mean-field accuracy but also the predictive uncertainty of reduced-order surrogate models for CFD. We evaluated the approach on three benchmark cases that progress in application relevance up to a cerebral aneurysm. Across all benchmarks, the ensemble correctly predicts core flow features and provides uncertainty estimates. Providing such uncertainty estimates is a necessary step toward using ROMs in clinical or other decision-centric contexts, where both accuracy and well-quantified confidence matter.

Mean-field accuracy For the unsteady advection-diffusion-reaction system, the ensemble reaches moderate errors ($\epsilon_{\text{rel}} = 4.3 \times 10^{-2}$, $\epsilon_{\max} = 1.13 \times 10^{-1}$), with residuals focused near the transported source and along strongly advective directions. In the von Kármán cylinder case, the model reconstructs velocity magnitude and pressure with low errors ($\epsilon_{\text{rel}} = 3.70 \times 10^{-2}$ and 4.59×10^{-2}; $\epsilon_{\max} = 8.3 \times 10^{-2}$ and 1.4×10^{-1}), degrading predictably at separation and within convecting vortex cores. In the three-dimensional cerebral-aneurysm case, errors are an order of magnitude smaller than in the other benchmarks ($\epsilon_{\text{rel}} = 0.34 \times 10^{-2}$ for velocity, 0.45×10^{-2} for pressure) with comparably low maxima.

At first glance, the high accuracy on the geometrically most complex case seems counterintuitive. However, the results matches the complexity of the parameter space and the resulting variety in flow across cases. Case 1 has a single scalar field but spans a 4D parameter space with broader variability, requiring the model to learn more complex flow patterns. Case 2 varies only one inflow parameter, but the flow dynamics are more complicated due to the vortex shedding. In case 3, we also vary a single inflow parameter but within a physiologic range that preserves flow patterns in and around the aneurysm and yields a smooth, low-dimensional latent trajectory. This means that the patterns that the model needs to predict are fairly similar across the parameter space. In both case 2 and case 3, velocity is predicted more accurately than pressure. This expected, since pressure fields exhibit an overall higher complexity and sharper changes in intensity, patterns which make them harder to learn for a neural network.

Compared to [18], the velocity field errors on case 2, the Von Kármán vortex street, are higher ($\epsilon_{\text{rel}} = 1.49 \times 10^{-2}$ and $\epsilon_{\max} = 2.53 \times 10^{-2}$ reported there). Pressure errors are not reported in [18]. The discrepancy most likely is a result of to differences in the FOM discretization. In contrast, we greatly exceed their accuracy on the aneurysm. Here, two factors likely contribute: (i) a simpler vascular geometry without complex bifurcations and (ii) a smoother inlet profile due to using a cosine inlet velocity as an approximation of a cardiac cycle, where [18] parameterized their inlet flow rate by interpolating radial basis functions to match the flow rate across the cardiac cycle). Both reduce the intrinsic dimensionality of the solution manifold and increase POD compressibility.

Predicted uncertainty Looking at the uncertainty quality, it becomes clear that there are differences between the cases. For the unsteady advection-diffusion-reaction system and the

Von Kármán vortex street, the predicted variances correspond well with the residuals, as clearly visible when looking at the sparsification plot and also indicated by the low areas under the sparsification error curves (AUSE = 0.091 for the scalar case; 0.0285 for velocity and 0.0492 for pressure in the cylinder). Across the two cases, higher variance correlates with regions of higher velocity or pressure magnitudes and more complex flow patterns, and larger residuals. This clearly shows that the variance is useful in highlighting areas in the flow that are more complicated, which are correspondingly the regions the model is not certain about. However, the calibration is only acceptable for the scalar case (CE = 0.077) and weaker for the cylinder (CE = 0.154 for velocity, 0.181 for pressure). This can also be clearly seen in the S-shaped calibration plot, indicating underconfidence for narrow and overconfidence for wide intervals.

The aneurysm shows a different pattern: very small mean errors but degraded AUSE and CE (AUSE = 0.39 velocity, 0.54 pressure; CE $\approx$ 0.18-0.19). With only $N_{\text{train}} = 6$ test cases and a narrow parameter range, the most likely reason for the badly calibrated predicted uncertainty is low ensemble diversity, meaning that all the models coverge towards a single value, and inherently small variances in the training data which can then also not be picked up by the heteroscedastic heads during training.

It can be concluded that high accuracy does not imply calibrated uncertainty. With sufficient variability during training, the quality of the predicted uncertainty seems to go up and make it more useful. But even for case 1 and 2, where the variability in the training data yields acceptable variances, they are not well-calibrated, clearly highlighting the need for calibration post-model training.

Performance The achieved per-case speedups range from 84× to 286× (Table 3.2). These values are lower than those reported in single-model POD-DL-ROM studies, where speedups on the order of 10^3–10^5 have been reported [18]. However, performance was not the focus of this work. The present architecture evaluates an ensemble of eight networks sequentially, entailing roughly eight times more forward passes and additional computations for predictive variance estimation and ensemble aggregation. Comparable performance to previously reported ranges could likely be achieved by optimizing the model with regard to runtime speed.

Limitations and future work The main limitations of this work stem from the case setup, the generation of training data, and the model formulation. While data-driven reduced-order models such as the POD-DL-ROM can learn highly accurate mappings from parameters to flow fields, they do not necessarily guarantee physical consistency. This limitation becomes most apparent for quantities such as pressure, where small deviations in the velocity field can accumulate into larger residuals in the momentum balance. Future work could therefore focus on integrating physics-informed losses into the POD-DL-ROM training objective, for instance by penalizing violations of the Navier-Stokes equations or the incompressibility constraint in the decoder output. Such a hybrid physics-informed POD-DL-ROM would preserve the computational efficiency of the reduced-order model while ensuring physically plausible reconstructions, similar in spirit to recent PINN formulations. Specifically interesting are PINN formulations that operate in the spirit of the POD-DL-ROM, meaning that the equations used for calculating the physics-based losses are stated in the low-dimensional POD space. This combination would not only enhance physical consistency and enable the model to predict in regions outside of the training parameter range with lower errors, but also keep model size small and training efficient.

A second limitation concerns the generation and variability of training data. The results demonstrate that both the amount and diversity of training samples strongly influence not only the mean-field accuracy but also the calibration of the predicted uncertainty. In particular, narrow parameter ranges lead to almost deterministic latent trajectories and limited diversity in the flow patterns, which in turn results in under-dispersed uncertainty estimates. Future studies should therefore explore more varied parameter sets and case designs, especially for clinically relevant scenarios such as the cerebral aneurysm. Introducing additional physiological parameters could help assess the robustness of the model and the expressiveness of the latent space under more realistic variability.

A further limitation arises from the geometric generality of the model. The current implementation is trained and evaluated on a single fixed mesh, which constrains its applicability to patient-specific anatomy. To enable deployment in clinical workflows, the model would need to incorporate geometric information as part of its input representation. One promising direction is to couple the POD-DL-ROM with statistical shape modeling of patient-specific vessel geometries, allowing the model to generalize across unseen shapes. Alternatively, using mesh-independent representations such as neural operators or graph-based encoders could further improve geometric flexibility. The present framework also neglects wall motion and fluid-structure interaction (FSI), which can play a major role in arterial hemodynamics. Extending the model to account for FSI effects would provide a more realistic representation of physiological flow dynamics and could enable predictive modeling of wall shear stress or aneurysm deformation over time.

Finally, when it comes to model uncertainty, we chose to train a deep ensemble with a heteroscedastic head to capture variance in predicted flow fields due to the relative ease of implementation and the solid scientific evidence across domains. However, the predicted uncertainty is not well-calibrated across any case and, as can be seen in the cerebral aneurysm case, is not even always guaranteed to be correlated with larger residuals. On this front, it would be interesting to explore newer approaches to uncertainty quantification. Furthermore, we assumed the CFD model to be the ground truth, ignoring clinical reality. However, there is of course also uncertainty associated with the training data we used for the construction of our surrogate model itself. It would therefore be very interesting to look into approaches of quantifying not only the uncertainty of the predictions of the surrogate model with regard to the original CFD prediction, but also with regards to real-world data.

Finally, when it comes to model uncertainty, we chose to train a deep ensemble with a heteroscedastic head to capture variance in the predicted flow fields, due to its relative ease of implementation and the strong empirical evidence supporting this approach across domains. However, the predicted uncertainty was not consistently well-calibrated across cases and, as seen in the cerebral aneurysm, was not always correlated with larger residuals. On this front, it would be interesting to explore approaches for post-hoc calibration or more expressive Bayesian methods that could provide better-calibrated uncertainty estimates. Moreover, it would be valuable to extend the present work towards a clearer separation between epistemic and aleatoric uncertainty, as the two are only implicitly captured by the ensemble and the heteroscedastic head in the current formulation. Finally, we assumed the CFD model to represent ground truth, ignoring uncertainties inherent to the underlying numerical model and to clinical reality. Future work could therefore aim to quantify not only the uncertainty of the surrogate with respect to the CFD simulations, but also with respect to real-world measurements.

Summary The POD-DL-ROM ensemble achieves accurate mean-field reconstructions across all benchmark cases and provides uncertainty estimates that correlate with residuals when sufficient variability is present in the training data. Compared to [18], the von Kármán velocity errors are higher, likely due to differences in the case setup yielding more complex vortex shedding, whereas the cerebral aneurysm results show substantially better accuracy, possibly due to a simpler geometry and smoother inlet velocity. While the ensemble predicts the cell-level uncertainty associated with the model's prediction, the quantitative assessment of the calibration of the uncertainty estimates indicates that they are not always trustful and highlights the importance of such measures. Improving model calibration, extending geometric generalization, and incorporating physical constraints represent direct next steps toward reliable ROMs for clinical decision support.

Bibliography

[1] J. H. Ferziger, M. Perić, and R. L. Street, *Computational methods for fluid dynamics.* springer, 2019.

[2] "Hyperbolic Problems", in *Numerical Approximation of Partial Differential Equations*, vol. 23, Springer Berlin Heidelberg, 1994, pp. 449–508. DOI: 10.1007/978-3-540-85268-1_14.

[3] P. R. Spalart and V. Venkatakrishnan, "On the role and challenges of CFD in the aerospace industry", *The Aeronautical Journal*, vol. 120, no. 1223, pp. 209–232, 2016. DOI: 10.1017/aer.2015.10.

[4] A. T. G. Tapeh and M. Z. Naser, "Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices", *Archives of Computational Methods in Engineering*, vol. 30, no. 1, pp. 115–159, 2023. DOI: 10.1007/s11831-022-09793-w.

[5] M.-P. Hosseini, A. Hosseini, and K. Ahi, "A Review on Machine Learning for EEG Signal Processing in Bioengineering", *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, 2021. DOI: 10.1109/RBME.2020.2969915.

[6] P. Benner, S. Gugercin, and K. Willcox, "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems", *SIAM Review*, vol. 57, no. 4, pp. 483–531, 2015. DOI: 10.1137/130932715.

[7] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira, "Volume 1 System- and Data-Driven Methods and Algorithms", in *Volume 1 System- and Data-Driven Methods and Algorithms*, De Gruyter, 2021.

[8] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction.* Springer, 2015.

[9] C. Farhat, S. Grimberg, A. Manzoni, A. Quarteroni, *et al.*, "Computational bottlenecks for proms: Precomputation and hyperreduction", *Model order reduction*, vol. 2, pp. 181–244, 2020.

[10] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery", *Drug Discovery Today*, vol. 23, no. 6, pp. 1241–1250, 2018. DOI: 10.1016/j.drudis.2018.01.039.

[11] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold", *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. DOI: 10.1038/s41586-021-03819-2.

[12] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, V. M. Tran, A. Chiappino-Pepe, A. H. Badran, I. W. Andrews, E. J. Chory, G. M. Church, E. D. Brown, T. S. Jaakkola, R. Barzilay, and J. J. Collins, "A Deep Learning Approach to Antibiotic Discovery", *Cell*, vol. 180, no. 4, 688–702.e13, 2020. DOI: 10.1016/j.cell.2020.01.021.

[13] M. Lino, S. Fotiadis, A. A. Bharath, and C. D. Cantwell, "Current and emerging deep-learning methods for the simulation of fluid dynamics", *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 479, no. 2275, p. 20 230 058, 2023. DOI: 10.1098/rspa.2023.0058.

[14] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations", *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. DOI: 10.1016/j.jcp.2018.10.045.

[15] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review", *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, 2021. DOI: 10.1007/s10409-021-01148-1.

[16] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators", *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, 2021. DOI: 10.1038/s42256-021-00302-5.

[17] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Fourier Neural Operator for Parametric Partial Differential Equations*, 2021. DOI: 10.48550/arXiv.2010.08895.

[18] S. Fresca and A. Manzoni, "POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition", *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114 181, 2022. DOI: 10.1016/j.cma.2021.114181.

[19] L. K. Hansen and P. Salamon, "Neural network ensembles", *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 2002.

[20] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review", *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105 151, 2022. DOI: 10.1016/j.engappai.2022.105151.

[21] B. Lakshminarayanan, A. Pritzel, and C. Blundell, *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*, 2017. DOI: 10.48550/arXiv.1612.01474.

[22] A. Renda, M. Barsacchi, A. Bechini, and F. Marcelloni, "Comparing ensemble strategies for deep learning: An application to facial expression recognition", *Expert Systems with Applications*, vol. 136, pp. 1–11, 2019. DOI: 10.1016/j.eswa.2019.06.025.

[23] A. Kendall and Y. Gal, *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?*, 2017. DOI: 10.48550/arXiv.1703.04977.

[24] A. Loquercio, M. Segu, and D. Scaramuzza, "A General Framework for Uncertainty Estimation in Deep Learning", *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020. DOI: 10.1109/LRA.2020.2974682.

[25] S. Seoni, V. Jahmunah, M. Salvi, P. D. Barua, F. Molinari, and U. R. Acharya, "Application of uncertainty quantification to artificial intelligence in healthcare: A review of last decade (2013–2023)", *Computers in Biology and Medicine*, vol. 165, p. 107 441, 2023. DOI: 10.1016/j.compbiomed.2023.107441.

[26] S. Razavi, A. Jakeman, A. Saltelli, C. Prieur, B. Iooss, E. Borgonovo, E. Plischke, S. Lo Piano, T. Iwanaga, W. Becker, S. Tarantola, J. H. Guillaume, J. Jakeman, H. Gupta, N. Melillo, G. Rabitti, V. Chabridon, Q. Duan, X. Sun, S. Smith, R. Sheikholeslami, N. Hosseini, M. Asadzadeh, A. Puy, S. Kucherenko, and H. R. Maier, "The Future of Sensitivity Analysis: An essential discipline for systems modeling and policy support", *Environmental Modelling & Software*, vol. 137, p. 104 954, 2021. DOI: 10.1016/j.envsoft.2020.104954.

[27] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, "Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons", *Journal of Computational Physics*, vol. 477, p. 111 902, 2023. DOI: 10.1016/j.jcp.2022.111902.

[28] E. Begoli, T. Bhattacharya, and D. Kusnezov, "The need for uncertainty quantification in machine-assisted medical decision making", *Nature Machine Intelligence*, vol. 1, no. 1, pp. 20–23, 2019. DOI: 10.1038/s42256-018-0004-1.

[29] M. Chu, *Uncertainty Quantification in Computational Fluid Dynamics: Physics and Machine Learning Based Approaches*, 2024. DOI: 10.48550/arXiv.2312.14684.

[30] J. Gast and S. Roth, "Lightweight Probabilistic Deep Networks", 2018, pp. 3369–3378.

[31] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, V. Makarenkov, and S. Nahavandi, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges", *Information Fusion*, vol. 76, pp. 243–297, 2021. DOI: 10.1016/j.inffus.2021.05.008.

[32] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012.

[33] A. Ganguly and S. W. F. Earp, *An Introduction to Variational Inference*, 2021. DOI: 10.48550/arXiv.2108.13083.

[34] A. Graves, "Practical variational inference for neural networks", *Advances in neural information processing systems*, vol. 24, 2011.

[35] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, *Weight Uncertainty in Neural Networks*, 2015. DOI: 10.48550/arXiv.1505.05424.

[36] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning", in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, pp. 1050–1059.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[38] J. Hron and A. Kendall, *Concrete Dropout*, 2017. DOI: 10.48550/arXiv.1705.07832.

[39] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift", in *Advances in Neural Information Processing Systems*, 2019.

[40] O. Sagi and L. Rokach, "Ensemble learning: A survey", *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 8, no. 4, e1249, 2018.

[41] N. Halko, P.-G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, 2010. DOI: 10.48550/arXiv.0909.4061.

[42] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.

[43] G. Cybenko, “Approximation by superpositions of a sigmoidal function”, *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[44] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators”, *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[45] K. O’Shea and R. Nash, *An Introduction to Convolutional Neural Networks*, 2015. DOI: 10.48550/arXiv.1511.08458.

[46] X. Guo, W. Li, and F. Iorio, “Convolutional Neural Networks for Steady Flow Approximation”, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 481–490. DOI: 10.1145/2939672.2939738.

[47] K. Lee and K. T. Carlberg, “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”, *Journal of Computational Physics*, vol. 404, p. 108 973, 2020. DOI: 10.1016/j.jcp.2019.108973.

[48] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks”, *Science*, vol. 313, no. 5786, pp. 504–507, 2006. DOI: 10.1126/science.1127647.

[49] U. Michelucci, *An Introduction to Autoencoders*, 2022. DOI: 10.48550/arXiv.2201.03898.

[50] E. Plaut, “From principal subspaces to principal components with linear autoencoders”, *arXiv preprint arXiv:1804.10253*, 2018.

[51] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction”, 2011. DOI: 10.1007/978-3-642-21735-7_7.

[52] S. Fresca, L. Dede’, and A. Manzoni, “A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs”, *Journal of Scientific Computing*, vol. 87, no. 2, p. 61, 2021. DOI: 10.1007/s10915-021-01462-7.

[53] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 1996.

[54] S. Kristoffersson Lind, Z. Xiong, P.-E. Forssén, and V. Krüger, “Uncertainty quantification metrics for deep regression”, *Pattern Recognition Letters*, vol. 186, pp. 91–97, 2024. DOI: 10.1016/j.patrec.2024.09.011.

[55] E. Ilg, Ö. Çiçek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, “Uncertainty Estimates and Multi-Hypotheses Networks for Optical Flow”, 2018.

[56] A. Niculescu-Mizil and R. Caruana, “Predicting good probabilities with supervised learning”, in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 625–632. DOI: 10.1145/1102351.1102430.

[57] M. P. Naeini, G. Cooper, and M. Hauskrecht, “Obtaining Well Calibrated Probabilities Using Bayesian Binning”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015. DOI: 10.1609/aaai.v29i1.9602.

[58] V. Kuleshov, N. Fenner, and S. Ermon, “Accurate Uncertainties for Deep Learning Using Calibrated Regression”, in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2796–2804.

[59] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques”, *Computers in physics*, vol. 12, no. 6, pp. 620–631, 1998.

[60] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch”, in *NIPS-W*, 2017.

[61] S. Fresca and A. Manzoni, “Real-Time Simulation of Parameter-Dependent Fluid Flows through Deep Learning-Based Reduced Order Models”, *Fluids*, vol. 6, no. 7, p. 259, 2021. DOI: 10.3390/fluids6070259.

[62] N. Chalouhi, B. L. Hoh, and D. Hasan, “Review of Cerebral Aneurysm Formation, Growth, and Rupture”, *Stroke*, vol. 44, no. 12, pp. 3613–3622, 2013. DOI: 10.1161/STROKEAHA.113.002390.

[63] J. R. Cebral, F. Mut, J. Weir, and C. M. Putman, “Association of Hemodynamic Characteristics and Cerebral Aneurysm Rupture”, *American Journal of Neuroradiology*, vol. 32, no. 2, pp. 264–270, 2011. DOI: 10.3174/ajnr.A2274.

[64] J. Cebral, F. Mut, J. Weir, and C. Putman, “Quantitative Characterization of the Hemodynamic Environment in Ruptured and Unruptured Brain Aneurysms”, *AJNR: American Journal of Neuroradiology*, vol. 32, no. 1, pp. 145–151, 2011. DOI: 10.3174/ajnr.A2419.

[65] M. Shojima, M. Oshima, K. Takagi, R. Torii, M. Hayakawa, K. Katada, A. Morita, and T. Kirino, “Magnitude and Role of Wall Shear Stress on Cerebral Aneurysm”, *Stroke*, vol. 35, no. 11, pp. 2500–2505, 2004. DOI: 10.1161/01.STR.0000144648.89172.0f.

[66] N. Juchler, P. Bijlenga, and S. Hirsch, *AneuX morphology database*, 2022. DOI: 10.5281/zenodo.6678442.