Progressive multi-fidelity learning for physical system predictions

Paolo Conti[a,d,*], Mengwu Guo[b], Attilio Frangi[c], Andrea Manzoni[d]

[a] *The Alan Turing Institute, London, NW1 2DB, UK*
[b] *Centre for Mathematical Sciences, Lund University, Sweden*
[c] *Department of Civil and Environmental Engineering, Politecnico di Milano, Italy*
[d] *MOX – Department of Mathematics, Politecnico di Milano, Italy*

Abstract

Highly accurate datasets from numerical or physical experiments are often expensive and time-consuming to acquire, posing a significant challenge for applications that require precise evaluations, potentially across multiple scenarios and in real-time. Even building sufficiently accurate surrogate models can be extremely challenging with limited high-fidelity data. Conversely, less expensive, low-fidelity data can be computed more easily and encompass a broader range of scenarios. By leveraging multi-fidelity information, prediction capabilities of surrogates can be improved. However, in practical situations, data may be different in types, come from sources of different modalities, and not be concurrently available, further complicating the modeling process. To address these challenges, we introduce a progressive multi-fidelity surrogate model. This model can sequentially incorporate diverse data types using tailored encoders. Multi-fidelity regression from the encoded inputs to the target quantities of interest is then performed using neural networks. Input information progressively flows from lower to higher fidelity levels through two sets of connections: *concatenations* among all the encoded inputs, and *additive connections* among the final outputs. This dual connection system enables the model to exploit correlations among different datasets while ensuring that each level makes an additive correction to the previous level without altering it. This approach prevents performance degradation as new input data are integrated into the model and automatically adapts predictions based on the available inputs. We demonstrate the effectiveness of the approach on numerical benchmarks and a real-world air pollution case study, showing that it reliably integrates multi-modal data, mitigates low-fidelity imperfections, and provides accurate predictions, while maintaining performance when generalizing across time and parameter variations.

Keywords: Multi-fidelity, data fusion, progressive learning, surrogate model, scientific machine learning

1. Introduction

Scientific computing today stands as a cornerstone in several fields, delivering unparalleled success in diverse applications in science and engineering, including weather forecast [1, 2, 3], computational biology [4], among many others. However, in spite of its undeniable success, the computational demands of high-fidelity simulations may become prohibitively expensive. This computational burden becomes critical in multi-query scenarios such as uncertainty quantification, optimal control, or model calibration, where repeated evaluations are required. In these challenging scenarios, the construction of efficient surrogate models thus emerges as a crucial strategy. Such models serve as indispensable proxies, enabling the cost-effective and accurate prediction of physical systems. In particular, data-driven surrogates are capable of learning complex system dynamics directly from high-dimensional datasets, without relying on explicit physical knowledge of the underlying process. By extracting rich low-dimensional latent representations and capturing nonlinear correlations, these models provide accurate reduced-order approximations, making them a powerful tool for efficient and scalable scientific computing [5, 6, 7, 8, 9, 10].

*Corresponding author.
Email addresses: pconti@turing.ac.uk (Paolo Conti), mengwu.guo@math.lu.se (Mengwu Guo), attilio.frangi@polimi.it (Attilio Frangi), andrea1.manzoni@polimi.it (Andrea Manzoni)

However, the effectiveness and trustworthiness of deep learning surrogates can deteriorate when acquiring high-fidelity data is prohibitively expensive to span the entire modeling domain adequately or even to fit a surrogate model. On the other hand, we might frequently rely on a wider range of low-fidelity data, obtained for instance from coarser simulations or simplified models, as well as heterogeneous sources of information. These additional datasets may differ in nature and modality, carrying complementary information: effectively integrating multiple data streams has the potential to improve predictive capabilities and foster a more holistic understanding of complex physical systems.

In these scenarios, multi-fidelity strategies emerge as a powerful solution in many areas of science and engineering [11, 12, 13, 14, 15]. By leveraging data of different fidelity levels, multi-fidelity approaches enhance the model's ability to generalize across different regions, particularly where high-fidelity information is limited or unavailable, thereby achieving an effective compromise between computational cost and model accuracy. A wide range of multi-fidelity surrogate modeling techniques are developed using Gaussian processes [16, 17, 18, 19], and, more recently, deep learning methods [20, 21, 22, 23, 24], which address key challenges such as high dimensionality, scalability, and strongly nonlinear correlations across datasets. Nevertheless, most existing approaches are tailored to settings where datasets are structurally homogeneous – for example, solution data generated with the same solver or model at varying discretization levels, or data from different solvers producing the same output quantities of interest [12]. Therefore, challenges arise in incorporating diverse sources of information or different data types, a problem often referred to in machine learning as *multi-modal* learning, impacting a wealth of application fields ranging from cancer biomarker discovery [25], precision medicine [26] and neuroimaging [27] to wearable sensors [28] and meteorological monitoring [29], just to mention a few. Additionally, datasets may not be available simultaneously at the time of surrogate construction but acquired at different stages, in which case retraining neural networks with new inputs may lead to the issue known as *catastrophic forgetting*.

To tackle these challenges, we propose a progressive multi-fidelity surrogate modeling paradigm, designed to incorporate new datasets of varying fidelity as they become available. This allows to progressively enhance prediction accuracy and reduce uncertainty, while effectively ensuring knowledge retention across updates. This model can sequentially incorporate diverse data types using tailored encoder neural networks. The encoders process and transform data into a meaningful latent representation that is suitable to be merged with the other fidelity datasets. The architecture of each encoder is selected according to the data modality: for example, feedforward networks for vector-valued data, recurrent networks (e.g., LSTMs [30]) for time series, convolutional networks [31] or vision transformers [32] for image data, and so on. The encoded features are concatenated and mapped to the target quantities of interest through a decoder, which is trained in a supervised manner on high-fidelity data.

Fig. 1 illustrates how input information flows from lower to higher fidelity levels through two types of connections: *concatenations* among all encoded inputs and *additive connections* among the decoded outputs. This dual connection system enables exploiting correlations among different datasets while ensuring that each level makes an additive correction to the previous level without altering it. This approach protects against catastrophic forgetting and performance degradation as new input data is integrated into the model.

Moreover, this progressive multi-fidelity structure is beneficial not only in training but also in online deployment, as it adapts predictions to the inputs available at runtime, ensuring operational continuity. When higher-fidelity input data are unavailable due to time, budget, or sensor constraints, the model can seamlessly fall back to lower-fidelity levels. By leveraging the inductive bias of the progressive structure and the information contained in physically meaningful low-fidelity data, the framework mitigates the lack of physical consistency typical of data-driven methods in data-scarce regimes, thereby enhancing accuracy in generalization across time and parameters and improving reliability.

To evaluate the framework, we consider three applications spanning both *(i)* synthetic data obtained through the solution of nonlinear, time-dependent partial differential equations (PDEs) and *(ii)* real-world data. First, a reaction-diffusion system reconstructs high-fidelity spiral wave dynamics from coarse, noisy low-fidelity simulations. Second, a Navier–Stokes benchmark reconstructs vortex shedding from hierarchical low-fidelity inputs, including drag/lift coefficients, outlet sensors, and partial-domain snapshots. Third, an air pollution scenario estimates benzene concentrations from low-cost sensor data despite missing or unreliable inputs. These cases highlight key challenges: integrating multi-modal data, mitigating low-fidelity imperfections, and enabling reliable temporal and parametric extrapolation.

Our method builds upon the progressive networks proposed by [33], which employ lateral connections

to accumulate knowledge across reinforcement learning tasks. We generalize this concept to the context of multi-fidelity and multi-modal learning for physical systems. Related approaches include multi-fidelity neural processes [34], which hierarchically integrate datasets but do not incorporate additive corrections to prevent catastrophic forgetting, and multi-fidelity reduced-order surrogate models [12], which are limited to two fidelity levels of homogeneous modality.

In summary, our contributions include:

- A progressive multi-fidelity architecture that sequentially integrates heterogeneous datasets while preserving prior knowledge through additive corrections.
- A dual fusion strategy that integrates multi-fidelity data both at the latent level (early fusion of encoded features) and at the output level (late fusion via output corrections), enabling effective exploitation of complementary information across modalities.
- Extensive validation across a range of PDE benchmarks with diverse quantities of interest, as well as a real-world air pollution case study, demonstrating the great flexibility of the proposed architecture and practical relevance in cases where noisy data are acquired from real sensor measurements.

The paper is structured as follows. In Section 2, we detail the structure of the progressive multi-fidelity learning model and describe the procedures for offline training and online deployment. Sections 3–4–5 present and discuss the model's performance on the three applications introduced earlier, with conclusions drawn in Section 6. The source code of the proposed method, along with the datasets used in this work, is available in the public repository `https://github.com/ContiPaolo/Progressive-Multifidelity-NNs`.

2. Method

In this section, we present the method for constructing the progressive multi-fidelity surrogate model for physical system predictions. We assume that the low-fidelity input data can be arranged in a hierarchy, and, for simplicity, that there is a single high-fidelity output to be predicted. In the following, we detail the offline training and the online deployment phases of the method.

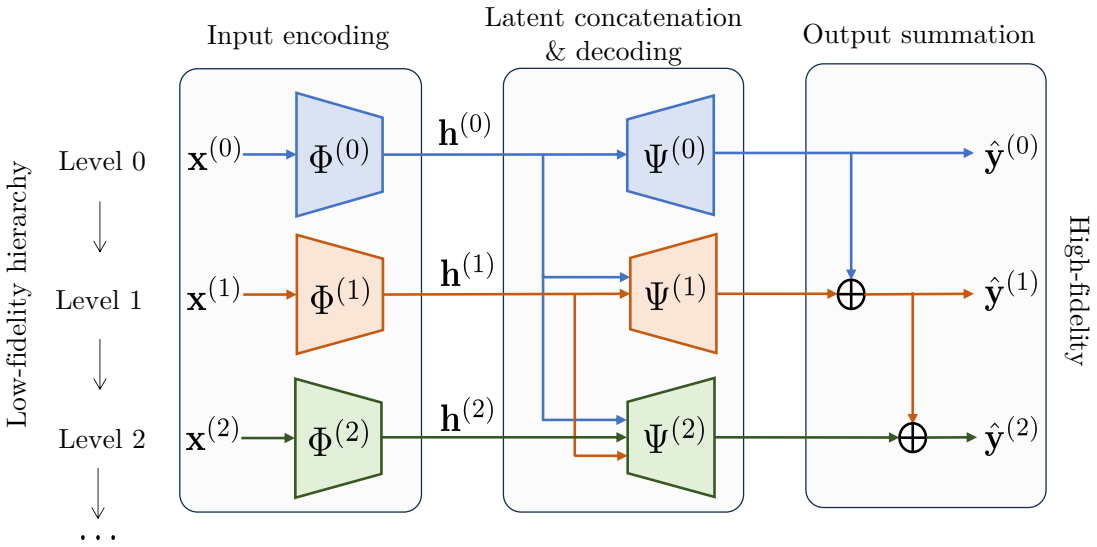

Figure 1: Progressive multi-fidelity surrogate modeling structure. The method is composed of three steps: (i) *input encoding*, at each level l, input data $\mathbf{x}^{(l)}$ are mapped by encoders $\Phi^{(l)}$ into latent variables $\mathbf{h}^{(l)}$; (ii) *latent concatenation and decoding*, latent variables are concatenated and decoded by $\Psi^{(l)}$; (iii) *output summation*, previous output is summed to the current one to form the final output $\hat{\mathbf{y}}^{(l)}$.

2.1. Offline training

The progressive multi-fidelity surrogate model is constructed through sequential levels, where at each level, a different low-fidelity dataset is provided as input, following the hierarchy. Each level is characterized by three steps: input encoding, decoding of latent concatenation, and output summation. Figure 1 presents a schematic overview of the approach.

- **Input encoding**. Since data sources can be of any kind, inputs are processed by an encoder neural network and transformed into a latent representation that is suitable to be merged with the other fidelity datasets. The choice of the encoder network architecture reflects the input data type. For instance, standard feedforward (FF) neural networks (NNs) can be employed with parameter values and vector-valued data; recurrent NNs, such as long short-term memory (LSTM) networks, for time series; convolutional NNs or vision transformers for image data. Hence, at each level l, the input data $\mathbf{x}^{(l)} \in \mathbb{R}^{d_{\text{in}}^{(l)}}$ are passed to an encoder network $\Phi^{(l)}$:

$$\Phi^{(l)}(\cdot\,; \mathbf{W}_{\Phi}^{(l)}) : \mathbb{R}^{d_{\text{in}}^{(l)}} \to \mathbb{R}^{d_{\text{h}}^{(l)}}, \text{ such that } \mathbf{x}^{(l)} \mapsto \Phi^{(l)}(\mathbf{x}^{(l)}) =: \mathbf{h}^{(l)}, \tag{1}$$

  where $\mathbf{h}^{(l)}$ denotes the latent variable of dimension $d_{\text{h}}^{(l)}$, $\mathbf{W}_{\Phi}^{(l)}$ the network parameters and $d_{\text{in}}^{(l)}$ indicates the original dimension of the input data. The data encoding process is crucial for extracting data features and providing low-dimensional latent representation, especially when dealing with high-dimensional data by considering $d_{\text{h}}^{(l)} \ll d_{\text{in}}^{(l)}$. As data types might be different across different levels, the encoding process is specific to each level and independent from the others. Moreover, it is not necessary to use a neural network as the encoder; other encoding techniques allowing for dimensionality reduction or change of coordinates, such as Proper Orthogonal Decomposition (POD), can also be considered, as already proposed in [12].

- **Latent concatenation and decoding**. The current latent variable $\mathbf{h}^{(l)}$ is concatenated with all latent variables from previous levels $\{\mathbf{h}^{(j)}\}_{j<l}$ and processed by a decoder neural network. The role of the decoder is to map the concatenated latent variables into the output quantity of interest we aim to estimate. We expect the decoder to merge all fidelity information and exploit the correlations among all datasets. The type of the decoder network reflects the output data type and it remains the same across all levels, as each level aims to estimate the same high-fidelity output of interest. For instance, in physical systems governed by PDEs, spatio-temporal decoders are employed to reconstruct the evolution in time of the solution field. The decoder at level l is denoted as $\Psi^{(l)}$ and defined as:

$$\Psi^{(l)}(\cdot\,; \mathbf{W}_{\Psi}^{(l)}) : \mathbb{R}^{d_{\text{h}_{\text{tot}}}^{(l)}} \to \mathbb{R}^{d_{\text{out}}}, \text{ such that } \mathbf{h}_{\text{tot}}^{(l)} \mapsto \Psi^{(l)}(\mathbf{h}_{\text{tot}}^{(l)}), \tag{2}$$

  where $\mathbf{h}_{\text{tot}}^{(l)} := [\mathbf{h}^{(0)} \cdots \mathbf{h}^{(l-1)} \mathbf{h}^{(l)}] \in \mathbb{R}^{d_{\text{h}_{\text{tot}}}^{(l)}}$ with $d_{\text{h}_{\text{tot}}}^{(l)} = \sum_{j \le l} d_{\text{h}}^{(j)}$, which represents the dimension of the concatenated latent variables that serve as input to the decoder. $\mathbf{W}_{\Psi}^{(l)}$ are the decoder network parameters. The output dimension of the decoder d_{out} is the same at every level. Note that the present framework can be straightforwardly extended for the prediction of multiple outputs by considering multiple decoders.

- **Output summation**. In order to preserve information across the different levels, the output of the decoder is summed to the predicted output at the preceding level. Except for the first level, where the final output simply coincides with the decoder output $\hat{\mathbf{y}}^{(0)} = \Psi^{(0)}(\mathbf{h}^{(0)})$, for all $l > 0$ the final output is defined as

$$\hat{\mathbf{y}}^{(l)} = \hat{\mathbf{y}}^{(l-1)} + \Psi^{(l)}(\mathbf{h}_{\text{tot}}^{(l)}) \tag{3}$$

  In this way, the current level contributes an additive correction approximating the residual between the ground truth and the previous estimate, while avoiding the loss of information contained in predictions from earlier levels.

The three steps outlined above define how input data are processed, at each level, to produce a set of progressively refined estimates $\{\hat{\mathbf{y}}^{(l)}\}_{l=0:L}$, approximating the high-fidelity target quantities of interest $\mathbf{y}$. At the l-th level, the optimal parameters $\mathbf{W}_{\Phi}^{(j)}$ and $\mathbf{W}_{\Psi}^{(j)}$ of the neural networks for encoding and decoding, $\Phi^{(l)}$

and $\Psi^{(l)}$ respectively, are obtained by solving the following optimization problem through back-propagation with ADAM [35] algorithm:

$$\min_{\mathbf{W}_\Phi^{(j)}, \mathbf{W}_\Psi^{(j)}} \frac{1}{N} \sum_{n=1}^{N} \left\| \hat{\mathbf{y}}_n^{(l)} - \mathbf{y}_n^{(l)} \right\|_2^2 + \lambda_{\text{reg}} \left(\mathbf{W}_\Phi^{(j)}, \mathbf{W}_\Psi^{(j)} \right), \tag{4}$$

where $\hat{\mathbf{y}}_n^{(l)} = \hat{\mathbf{y}}_n^{(l)} \left(\{\mathbf{x}_n^{(j)}\}_{j \le l}; \{\mathbf{W}_\Phi^{(j)}, \mathbf{W}_\Psi^{(j)}\}_{j \le l} \right)$, and $n = 1, \ldots, N$ are the training data samples.

This loss function is defined as the mean squared error (MSE) between the predicted output at level l and the high-fidelity data, augmented with an additional regularization term denoted as λ_{reg}, intended to mitigate overfitting phenomena. Specifically, L_2 regularization is employed, i.e., the regularization term can be expressed as

$$\lambda_{\text{reg}} \left(\mathbf{W}_\Phi^{(j)}, \mathbf{W}_\Psi^{(j)} \right) = \lambda_\Phi \left\| \mathbf{W}_\Phi^{(j)} \right\|_2^2 + \lambda_\Psi \left\| \mathbf{W}_\Psi^{(j)} \right\|_2^2, \qquad \text{with } \lambda_\Phi, \lambda_\Psi \in \mathbb{R}^+. \tag{5}$$

The different levels are trained sequentially, following the hierarchy order. In particular, at the l-th level, the optimization process described in equation (4) determines only the current encoder and decoder networks, while the network weights $\{\mathbf{W}_\Phi^{(j)}, \mathbf{W}_\Psi^{(j)}\}_{j<l}$ remain fixed. This practice ensures that at each new level, the structure at lower levels remains unaltered during the optimization process, thus preventing the occurrence of catastrophic forgetting, which could otherwise degrade the model's performance at previous hierarchical levels.

The preceding levels convey information to the subsequent ones through both the concatenation of latent variables and the additive feedback from the previous output. Consequently, each new level configures itself as a correction to be applied to the previous level, which, in contrast, remains unchanged.

To ensure robustness and reliability of the predictions, the proposed method is equipped with uncertainty quantification. In this work, uncertainty in the model predictions is estimated using an ensemble-based technique [36], where the network is retrained multiple times with randomly initialized weights, and the statistical moments of the resulting ensemble of predictions are computed. As a potential future development, the decoder could be replaced by neural network architectures that inherently provide uncertainty estimates in a more computationally efficient manner, such as Bayesian NNs [37], Deep Kernel Learning methods [38], or Conditional Neural Processes [39].

2.2. Online prediction

Once the multi-fidelity networks are trained offline, they can be used to estimate high-fidelity outputs from the available low-fidelity inputs during the online prediction phase. The availability of low-fidelity input data depends on the online cost of assimilating or computing those data, given a certain computational budget or a time frame in which the progressive multi-fidelity model needs to operate. Therefore, if at testing time the budget allows for acquiring low-fidelity inputs $\{\mathbf{x}^{(l)}\}_{l \le \bar{l}}$ up to level $\bar{l}$, the model will produce the corresponding prediction $\hat{\mathbf{y}}^{(\bar{l})}$. The online deployment procedure is summarized in Algorithm 1. Note that

Algorithm 1 Online Deployment of Progressive Multi-Fidelity Surrogate Model

Input: Trained encoders $\{\Phi^{(j)}\}_{j=0}^{L}$, trained decoders $\{\Psi^{(j)}\}_{j=0}^{L}$, available low-fidelity inputs $\{\mathbf{x}^{(j)}\}_{j=0}^{\bar{l}}$ up to level $\bar{l} \le L$
Output: Predicted high-fidelity output $\hat{\mathbf{y}}^{(\bar{l})}$
Initialize prediction: $\hat{\mathbf{y}}^{(-1)} \leftarrow \mathbf{0}$
Initialize latent concatenation: $\mathbf{h}_{\text{tot}}^{(-1)} \leftarrow \emptyset$
for $l = 0$ **to** $\bar{l}$ **do**
    Encode input: $\mathbf{h}^{(l)} \leftarrow \Phi^{(l)}(\mathbf{x}^{(l)})$
    Concatenate latents: $\mathbf{h}_{\text{tot}}^{(l)} \leftarrow [\mathbf{h}_{\text{tot}}^{(l-1)}; \mathbf{h}^{(l)}]$
    Decode and sum outputs: $\hat{\mathbf{y}}^{(l)} \leftarrow \hat{\mathbf{y}}^{(l-1)} + \Psi^{(l)}(\mathbf{h}_{\text{tot}}^{(l)})$
end for
Return: $\hat{\mathbf{y}}^{(\bar{l})}$

the level $\bar{l}$ may vary over time, allowing the model to navigate the hierarchy of outputs and ensure continuity in the prediction of the corresponding high-fidelity quantities.

The hierarchical ordering here is based on the *availability* of input data rather than on data accuracy, as in traditional multi-level or multi-grid methods, where all data come from a single source and the hierarchy is determined by a control parameter balancing accuracy and computational cost. In our framework, data may come from multiple sources or modalities, and the hierarchy is defined solely by the availability of the data at the prediction stage [40].

2.3. Case studies

To demonstrate the versatility and robustness of our progressive multi-fidelity framework, we present three case studies spanning both synthetic and real-world scenarios:

- *Reaction–diffusion PDE problem.* A parametric, spatio-temporal system where high-fidelity simulations of spiral wave dynamics are reconstructed from coarse, noisy low-fidelity simulations with parametric bias.
- *Navier–Stokes PDE problem.* A computational fluid dynamics benchmark leveraging hierarchical low-fidelity inputs (drag and lift coefficients, outlet sensors, and partial-domain snapshots) to reconstruct unsteady flow behavior.
- *Air pollution monitoring.* A real-world case using sensor data that combine temperature, humidity, and co-pollutant measurements from low-cost devices to estimate expensive benzene concentrations, despite missing or unreliable low-fidelity signals.

These examples address key challenges in scientific machine learning: integrating multi-modal data (full-field simulations, boundary sensors, and scalar time series), mitigating low-fidelity imperfections (noise, model bias, and sparse coverage), and enabling reliable extrapolation beyond training regimes (temporal forecasting and parameter-space generalization). Across all cases, we quantify how each fidelity level improves predictions, balancing accuracy against computational or experimental costs. Uncertainty quantification highlights regimes where low-fidelity inputs are insufficient – critical for real-world deployment.

3. Application to simulation data (I): a nonlinear Reaction–Diffusion system

In this example, we aim to construct a surrogate model capable of generating spatio-temporal solutions of a nonlinear reaction–diffusion system, governed by the following partial differential equations:

$$\begin{aligned} \dot{u} &= \left(1 - \left(u^2 + v^2\right)\right) u + \mu \left(u^2 + v^2\right) v + D \left(u_{xx} + u_{yy}\right), \\ \dot{v} &= -\mu \left(u^2 + v^2\right) u + \left(1 - \left(u^2 + v^2\right)\right) v + D \left(v_{xx} + v_{yy}\right). \end{aligned} \tag{6}$$

The solution $[u, v]^T(x, y, t)$ represents two oscillating modes that generate spiral waves. The system (6) is defined over a spatial domain $(x, y) \in [-L, L]^2$ for $L = 10$ and a time span $t \in [0, T]$ for $T = 80$, where μ and D are parameters that respectively regulate the reaction and diffusion behaviors of the system. We prescribe periodic boundary conditions, and the initial condition is defined as

$$u(x, y, 0) = v(x, y, 0) = \tanh\left(\sqrt{x^2 + y^2} \cos\left((x + iy) - \sqrt{x^2 + y^2}\right)\right).$$

Our goal is to approximate the time evolution of the solution component u as functions of the reaction parameter $\mu \in \mathcal{P} = [0.5, 1.5]$ with a fixed diffusion coefficient $D = 0.05$.

3.1. Multi-fidelity setting

We aim to construct a multi-fidelity surrogate to efficiently evaluate high-resolution solution u over the whole spatio-temporal domain, as the reaction parameter μ varies, exploiting cheaply obtained, low-fidelity output quantities. In particular, we consider a multi-fidelity surrogate structured in three levels as illustrated in Fig. 2. All levels are trained to output the HF solutions, while they differ in the quality and modality of the data provided as input:

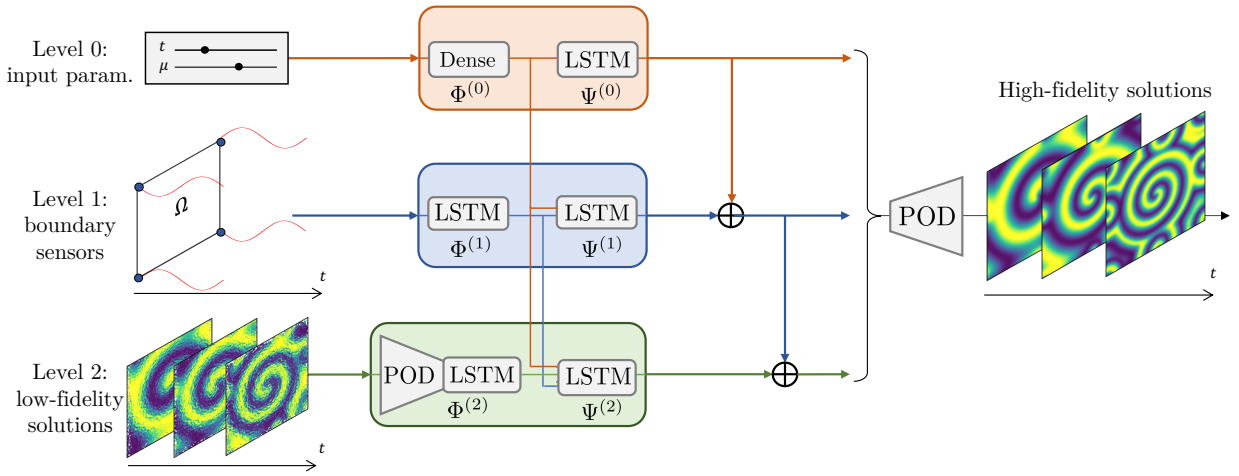

Figure 2: Multi-fidelity model for generating high-fidelity solutions to a reaction-diffusion system. At level 0 a deep-learning surrogate is constructed to estimate spatio-temporal solutions from the control parameters (time and reaction coefficient). In subsequent levels, information from artificial boundary sensors and cheap, low-resolution solutions are progressively integrated into the model. Encoder networks $\{\Phi^{(i)}\}_{i=0}^{2}$ process data from different modalities in a compatible latent representations. The decoder networks $\{\Psi^{(i)}\}_{i=0}^{2}$ at each level reconstruct spatio-temporal solutions from the latent representations by means of LSTM networks and POD.

- **Level 0 (input parameters).** A dense FF encoder network $\Phi^{(0)}$ is trained by providing as input data time t and diffusion parameter μ. No information about the solution is given as input.
- **Level 1 (boundary sensors).** Next level of the surrogate is trained on the temporal evolution of the solution at the domain's vertices. These data artificially simulate information from sensors placed at the domain's boundary. As we need to process time series data, we employ as encoder an LSTM neural network $\Phi^{(1)}$.
- **Level 2 (LF full-field solutions).** At this level, we exploit the whole LF information, i.e. we input the LF solutions over the entire spatial domain. As full-field, time series data are given as input, the encoder $\Phi^{(2)}$ needs to extract the dominant spatial patterns as well as encode temporal features. To this end, we combine proper orthogonal decomposition (POD) with LSTM networks. LF data are used to construct a LF POD basis onto which they are projected. The resulting time dependent expansion coefficients are then encoded by an LSTM network.

As the output we aim to predict is the spatio-temporal HF solutions, the decoder $\{\Psi^{(i)}\}_{i=0}^{2}$ combine temporal and spatial encoding, using LSTM networks and POD, respectively – mirroring the encoder structure of Level 2. Thus each level is trained to predict the HF POD coordinates, from which the entire spatial solution can be reconstructed. Note that the POD basis employed in the decoders is computed from HF snapshots, thus different from the LF POD basis in $\Phi^{(2)}$.

In this application, the quality of the input information determines the hierarchical order. Note that each level processes a dataset of different modality (vector valued, time-series, spatio-temporal snapshot data) through encoders tailored to the input data fusing their latent representation.

Dataset. Data are computed numerically by solving the PDEs (6) using the Fourier spectral method [41] with time step $\Delta t = 0.05$. Two distinct datasets are generated for use as input and output to the MF model, respectively, and are referred to as LF and HF based on the following characteristics:

- LF solution data are generated on a coarse equispaced spatial grid with $n_{\mathtt{LF}} = 32$ points in each direction, while a fine grid with $n_{\mathtt{HF}} = 100$ is adopted for the HF data. This reduces computational costs and allows LF data to be calculated much faster, at the price of worse accuracy.

- LF solutions are evaluated at a corrupted diffusion coefficient $D_{\texttt{LF}} = 0.1$, instead of $D_{\texttt{HF}} = D = 0.05$. This represents a bias in the LF modeling in terms of the physical property of viscosity and is employed to emulate model error.

- LF data are additionally contaminated by lognormal multiplicative noise with distribution $\mathcal{LN}(0, 0.8)$ to simulate observation error.

To train offline the model, a small amount of expensive HF solution data are computed over a shorter time horizon $T_{\text{train}} = 40 < T = 80$, at a limited set of parameter locations. In particular, $N_\mu = 10$ μ-values are selected over an equispaced grid of $\mathcal{P}$. We test the progressive MF method accuracy in predicting the HF solutions over an unseen set of parameters in $\mathcal{P}$ over the whole time window $[0, T]$. The input dimensions of the data at different fidelity levels are $d_{\text{in}}^{(0)} = 2$, $d_{\text{in}}^{(1)} = 4$, and $d_{\text{in}}^{(2)} = 1024$, corresponding to the time and parameter values at Level 0, four vertex sensors at Level 1, and the low-fidelity spatial degrees of freedom at Level 2. The output dimension of the high-fidelity solution is $d_{\text{out}} = 10000$. The dimensionality of both the Level 2 input and the high-fidelity output data is reduced using POD, retaining 9 and 11 modes, respectively, capturing about 90% of the total variance. Uncertainty bounds are computed from the statistical moments of trajectories obtained using ensemble-based methods over $m = 30$ retrainings, each with randomly initialized network weights. More details about the implementation are provided in Appendix A.

3.2. Results

For unseen testing values of the reaction coefficients $\mu \in \{0.875, 1.375\}$ and for different time instances $t \in \{20, 50, 70\}$, we show in Fig. 3 the reconstructed HF solution fields predicted by the progressive MF model at the different levels, along with the corresponding absolute errors. We notice a progressive improvement of the performance of the surrogate predictions with corresponding reduction in the error as we move up the hierarchy of the surrogate.

At level 0, the predictions are generally inaccurate, as the model attempts to reconstruct the entire solution field solely from the control parameters (i.e., time t and reaction coefficient μ). This task is particularly challenging due to the nonlinearity and high dimensionality of the parameter-to-solution map in a supervised regression setting with limited training data. As a result, in a single-fidelity context where no additional (LF) information is available, the model fails to extrapolate to unseen temporal scenarios, leading to poor forecasting performance. However, for time instances within the temporal window covered during training (albeit for unseen μ), level 0 predictions remain acceptable, capturing the overall solution patterns despite some loss in accuracy (see results for $t = 20$ in Fig. 3).

At level 1, temporal evolution data of the solution at the four vertices of the spatial domain are provided as additional inputs. Although this information is limited to the domain boundaries, we observe that incorporating physical insight leads to a significant improvement in prediction accuracy.

Finally, at level 2, the MF framework integrates fast and inexpensive LF solutions – albeit coarse but defined over the entire spatial domain – together with the knowledge acquired at previous levels. This enables the model to predict highly accurate solutions of the physical system, showing excellent agreement with the HF reference and substantially reduced prediction errors. Accessing this level during the online inference phase requires executing the LF solver to provide the input solutions, and feasibility depends on time or budget constraints. In this example, the computational time required to run the full LF simulation is approximately 0.58s[1], after which the progressive MF surrogate reconstructs the HF resolution, yielding an online speed-up of about 37× compared to solving the HF system directly. Notably, the LF inputs are computed using a corrupted model affected by both parametric error (incorrect diffusion coefficient) and measurement noise (lognormal distribution), demonstrating the robustness of the neural network in mitigating both model and observational biases Table 1 provides a quantitative comparison of relative errors with respect to reference HF test solutions.

As discussed in Sect. 3.1, the decoders at each level estimate the time evolution of the HF POD coefficients, which are then spatially reconstructed by projecting back using the POD spatial basis. To illustrate the forecasting performance across different levels for unseen parameter values, Fig. 4 shows the temporal evolution of several POD coefficients predicted by the LSTM network of decoders.

[1]Evaluated on a workstation with an AMD Ryzen 9 5950X 16-core processor.

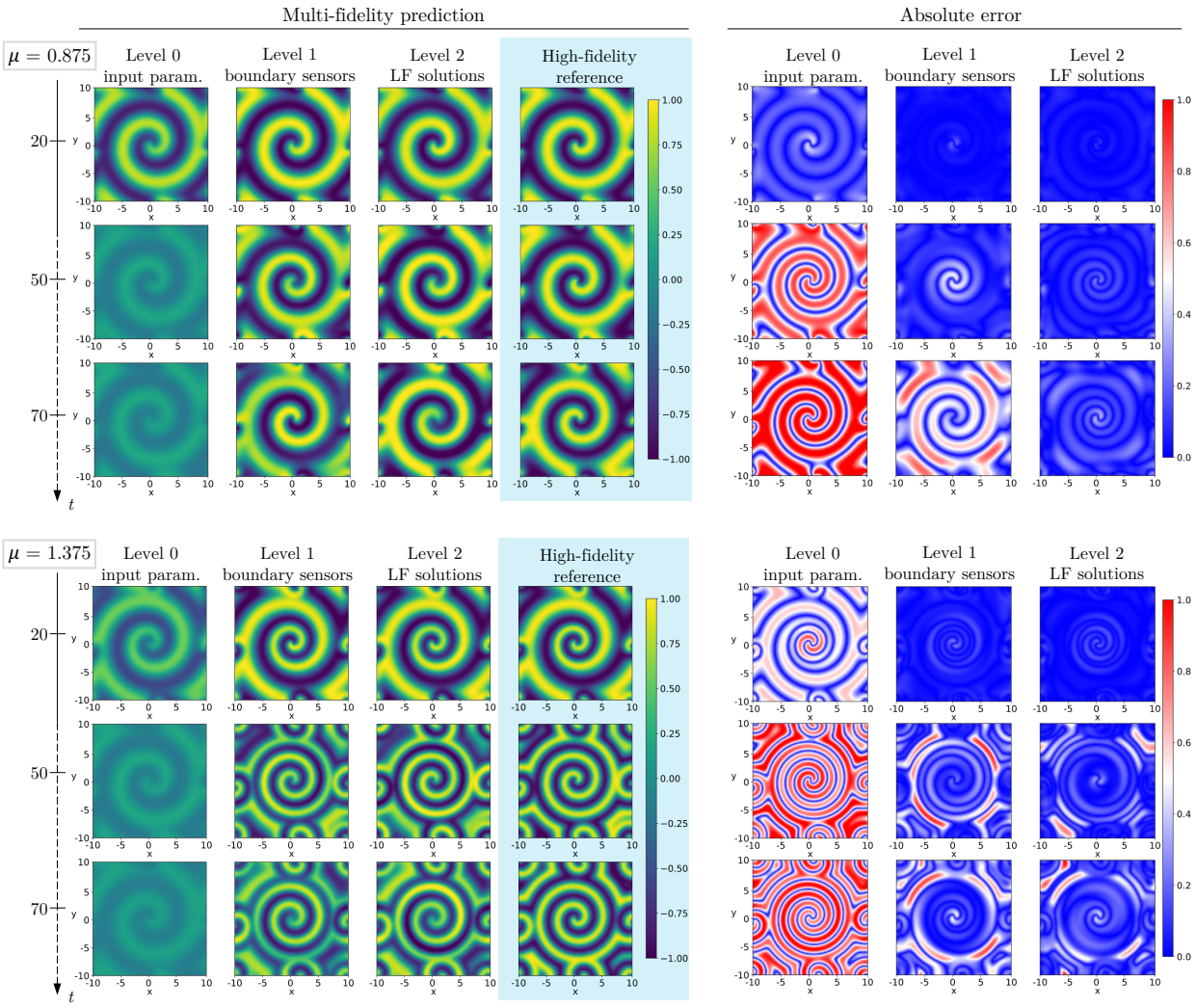

Figure 3: Progressive multi-fidelity prediction (left) for each fidelity level and absolute error (right) with respect to the high-fidelity reference solutions of the reaction-diffusion problem. The illustrated snapshots refer to one interpolated time instance $t = 20$ and two extrapolated time instances $t \in \{50, 70\}$ (being $T_{\text{train}} = 40$ the end of the HF training coverage) for two testing values of reaction parameter $\mu \in \{0.875, 1.375\}$ (top and bottom plots, respectively), unseen during the training.

We observe that predictions within the training time interval $[0, T_{\text{train}}]$, with $T_{\text{train}} = 40$, are accurate at all levels, highlighting the neural network's effectiveness as a surrogate for approximating high-dimensional, nonlinear systems in interpolation regimes – i.e., scenarios similar to those encountered during training. However, as we extrapolate beyond the training window into $[T_{\text{train}}, T] = [40, 80]$, the predictive performance at level 0 deteriorates significantly. This behavior is expected, given that this level relies solely on the control parameters as input, which do not convey any dynamical or physical information about the system. Nevertheless, the model at level 0 reflects this limitation by providing wide uncertainty bounds, reflecting the increased epistemic uncertainty in this regime. In contrast, levels 1 and 2 show a marked improvement in forecasting performance, accurately tracking the system dynamics over long-term horizons for unseen parameters. This clearly demonstrates how the integration of physically meaningful LF data helps mitigate the lack of physical consistency often observed in purely data-driven models, thus enabling robust extrapolation capabilities. As expected, transitioning from level 1 to level 2 results in further gains in predictive accuracy and a noticeable reduction in uncertainty. This improvement, however, comes at the cost of running LF solvers over the full spatial domain.

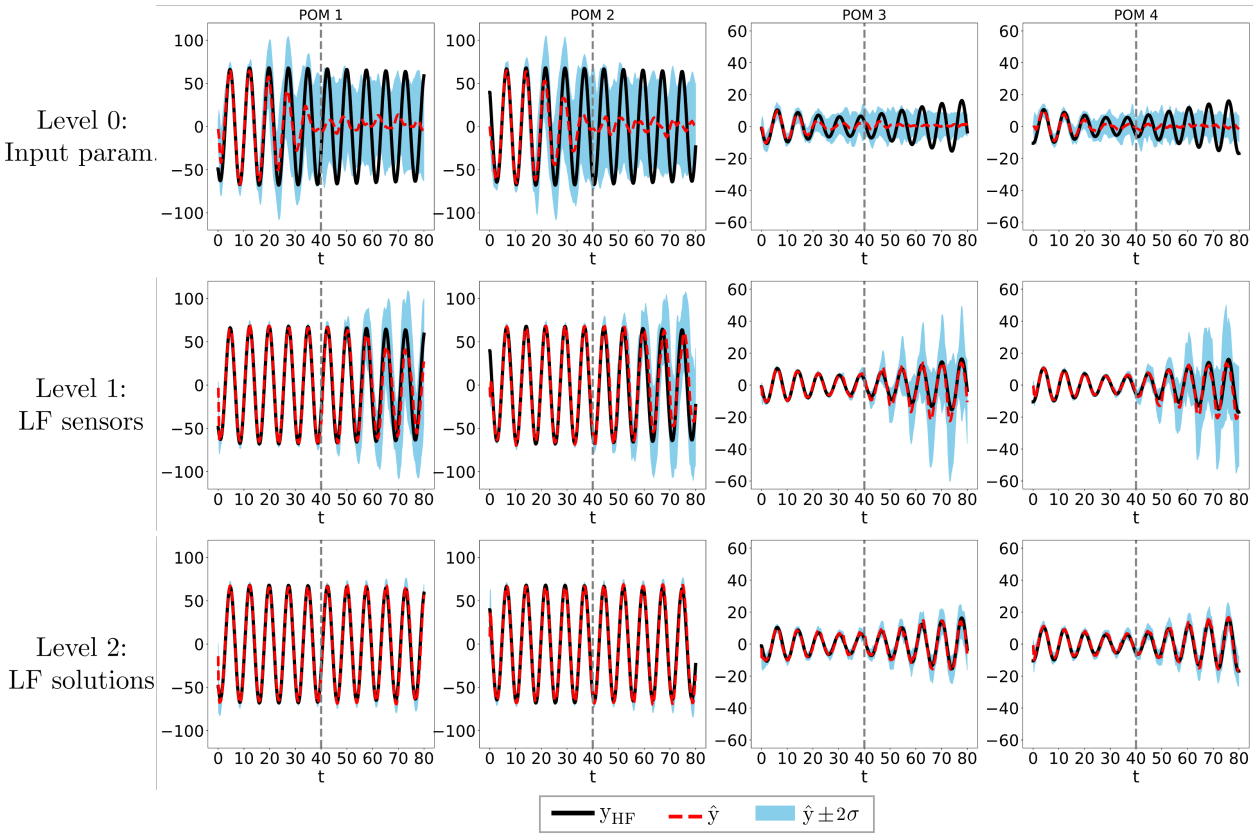

Figure 4: Comparison of the time evolution of the expansion coefficients of the POD modes (POM) in the reaction-diffusion example between the HF reference and the MF predicted solution. For the MF predictions, mean $\hat{y}$ and two standard deviation σ band over the different trainings are shown, providing uncertainty quantification. The vertical dashed line at $T_{\text{train}} = 40$ indicates the end time of HF data coverage in training, i.e., no HF information is available for $t \in [T_{\text{train}}, T]$.

4. Application to simulation data (II): Navier-Stokes fluid flows

We consider a two-dimensional fluid flow around a cylinder — a benchmark problem in computational fluid dynamics. For a viscous, incompressible Newtonian fluid flow, the problem is governed by the following Navier-Stokes equations

$$\begin{aligned} \rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{v}, p) &= \mathbf{0}, \qquad (\boldsymbol{x}, t) \in \Omega \times (0, T)\,, \\ \nabla \cdot \mathbf{v} &= 0, \qquad (\boldsymbol{x}, t) \in \Omega \times (0, T)\,, \end{aligned} \tag{7}$$

where $\mathbf{v}(\boldsymbol{x}, t)$ and $p(\boldsymbol{x}, t)$ represent the velocity and pressure field, respectively, and $\rho = 1.0$ kg/m^3 is the fluid density, and $\boldsymbol{\sigma}(\mathbf{v}, p) = -p\mathbf{I} + 2\nu\boldsymbol{\epsilon}(\mathbf{v})$ is the stress tensor, where $\boldsymbol{\epsilon}(\mathbf{v})$ denotes the strain tensor. The kinematic viscosity is defined as $\nu = 1/Re$, where the Reynolds number Re is the system parameter of interest. Here, the spatial domain $\Omega = (0, 2.2) \times (0, 0.41) \backslash B_r(0.2, 0.2)$ $(r = 0.05)$ represents a 2D channel with a cylindrical

Table 1: Relative error with respect to the HF reference solution, $\mathbf{y}$, is evaluated for the MF estimates, $\hat{\mathbf{y}}^{(l)}$, at each level l, as $\text{err}_{\%}^{(l)} = \frac{100\%}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \frac{\left\| \mathbf{y}(t_n, \mu_n) - \hat{\mathbf{y}}^{(l)}(t_n, \mu_n) \right\|_2}{\|\mathbf{y}(t_n, \mu_n)\|_2}$, where N_{test} is the number of time-parameter combinations in the test set.

	Level 0 (input param.)	Level 1 (boundary sensors)	Level 2 (LF solutions)
Relative error	81.3%	29.3%	18.6%

obstacle, where no-slip boundary conditions are prescribed on the side walls and on the cylinder, a parabolic inflow on the inlet, and an open boundary condition at the outlet. As the initial condition, we consider the fluid at rest.

In the present study, we consider $T = 20$s and $\mu = Re \in \mathcal{P} = [30, 100]$. When $Re < 49$, the flow exhibits a *steady* behavior; for larger Reynolds numbers, the flow transitions to an *unsteady* state, and a pair of vortices forms in the wake of the cylinder, oscillating periodically between top and bottom sides [42, 43].

4.1. Multi-fidelity setting

Our goal is to estimate the spatio-temporal evolution of the velocity magnitude $v(\boldsymbol{x}, t) = \|\mathbf{v}(\boldsymbol{x}, t)\|_2$ as a function of the Reynolds number, from the following low-fidelity information:

- **Level 0 (input parameters).** A FF encoder network $\Phi^{(0)}$ is trained by providing as input data time t and Reynolds number Re.
- **Level 1 (drag and lift).** The next level is trained on the drag and lift time series, two dimensionless, time-dependent coefficients that quantify forces acting on the obstacle [44]. Although they are informative on the system's temporal evolution, they are output quantities that do not provide a direct measure of the solution field. We employ an LSTM neural network $\Phi^{(1)}$ as encoder.
- **Level 2 (outflow).** Solution values at the outflow are provided as input, simulating sensors on the outlet boundary. The evolution of the outflow front measured at equispaced spatial points on the outlet is processed as a vector of time-series by an additional LSTM encoder $\Phi^{(2)}$.
- **Level 3 (partial domain).** This level simulates a video sensor that assimilates snapshots of the velocity magnitude over a partial domain surrounding the obstacle. Here, $\Phi^{(3)}$ employs an autoencoder architecture with LSTM layers to encode the high-dimensional spatio-temporal data.

To keep the training computationally light, we consider, as in the previous application, a combination of POD and LSTM networks as $\{\Psi^{(i)}\}_{i=0}^{3}$ to decode the spatio-temporal features of the high-fidelity velocity magnitude. A representation of the multi-fidelity architecture is shown in Fig. 5.

Dataset. Low-fidelity input and high-fidelity output data are generated through a finite element approximation of (7) with quadratic finite elements and backward differentiation formula provided by the MATLAB library `redbKIT` [45]. We consider a training set computed over a short time window $T_{\text{train}} = 15\text{s} < T = 20\text{s}$ at $N_\mu = 15$ Reynolds numbers equispaced over $\mathcal{P}$. High-fidelity data are computed over a mesh with 16478 triangular elements and $d_{\text{out}} = 8239$ vertices, while the temporal discretization is with step size $\Delta t = 5$ms. The high-fidelity output data consists of the time evolution of the velocity magnitude at the d_{out} spatial degrees of freedom (dof). The input dimensions of low-fidelity data are $d_{\text{in}}^{(0)} = 2, d_{\text{in}}^{(1)} = 2$ (time t and parameter μ at level 0, and drag and lift at level 1), and $d_{\text{in}}^{(2)} = 20, d_{\text{in}}^{(3)} = 577$ (spatial degrees of freedom on the outlet boundary and on a portion of the domain surrounding the obstacle; see Fig. 5). Further implementation details and training specifications are provided in Appendix A.

4.2. Results

For unseen Reynolds numbers $Re \in \{63, 92\}$ and different time instances $t \in \{9\text{s}, 19\text{s}\}$ (with $[0\text{s}, 15\text{s}]$ the training window), Fig. 6 shows the predicted high-fidelity velocity magnitude at each level of the progressive MF model, along with the corresponding absolute errors. At level 0, the surrogate relies solely on (t, Re) as inputs and can only reproduce the average stationary behavior of the flow, missing the unsteady vortex-shedding dynamics. Incorporating drag and lift coefficients at level 1 significantly improves predictions, despite the absence of any direct measurements of the solution field. Nonetheless, non-negligible errors remain in the wake region, where nonlinear effects dominate. At level 2, additional measurements from sensors on the outlet boundary enable accurate and spatially homogeneous predictions, even beyond the training time window. Notably, the accuracy at $t = 19$s is comparable to that at $t = 9$s, demonstrating the model's capability to reliably forecast forward in time.. Remarkably, improvements are observed throughout the domain – including the cylinder wake, where no sensors are placed – highlighting the model's ability to fuse and effectively exploit localized information. Finally, level 3, which assimilates partial-domain snapshots

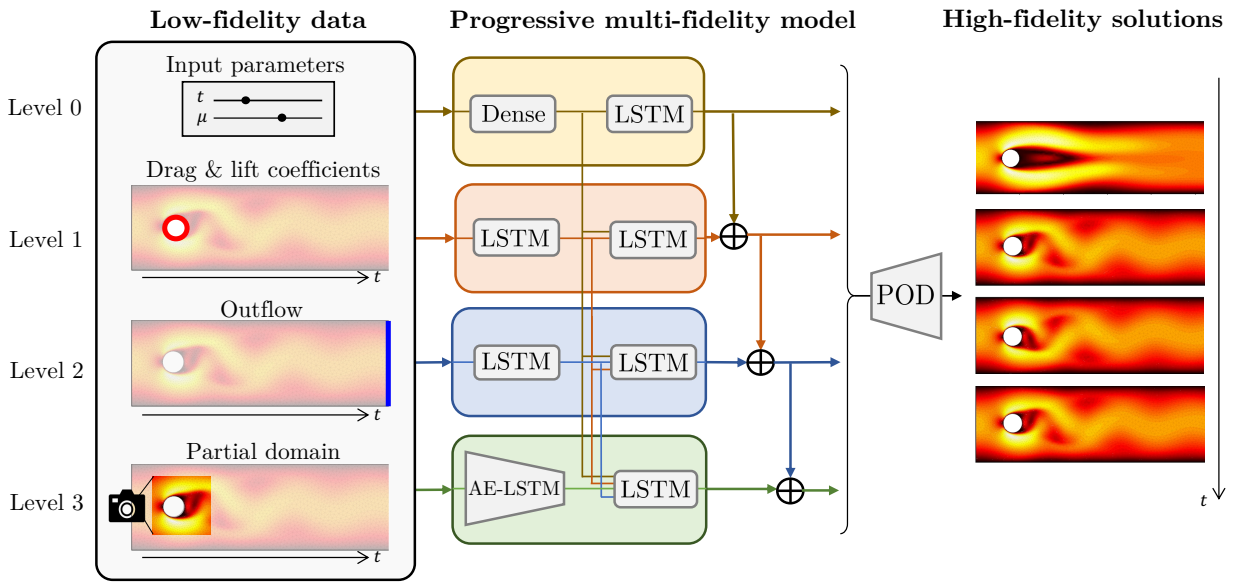

Figure 5: Multi-fidelity model for generating high-fidelity solutions to the Navier-Stokes equations. At level 0 a deep-learning surrogate is constructed to estimate spatio-temporal solutions from the input parameters (time and Reynolds number coefficient). At level 1 information about the time evolution of drag and lift coefficients, expressing the forces acting on the obstacle wall, are integrated. At level 2 information about solution outflow, representing boundary sensors, is included. Finally, spatio-temporal snapshots of a limited portion of the domain (surrounding the obstacle) are assimilated by the model at level 3. Encoder networks $\{\Phi^{(i)}\}_{i=0}^{3}$ process data from different modalities into compatible latent representations. The decoder networks $\{\Psi^{(i)}\}_{i=0}^{3}$ at each level reconstruct spatio-temporal solutions from the latent representation by means of LSTM networks and POD.

around the obstacle, yields only marginal gains over level 2, indicating that the outlet boundary data, combined with drag and lift measurements, already provide sufficiently rich information for the neural architecture. Table 4.2 reports the overall error over the full parametric, spatio-temporal test set, further supporting these conclusions.

5. Application to real measurement data: air pollution forecasting

Air pollution poses a critical threat to public health, requiring accurate monitoring of pollutant concentrations in urban areas [46]. Currently, air pollution monitoring is performed by fixed monitoring stations (based on industrial spectrometers), which can provide high-fidelity data with precise measurements but are expensive, bulky, and limited in spatial coverage, making evaluation in complex urban environments challenging. Low-cost multi-sensor devices, or "electronic noses", offer a complementary solution by increasing network density at a lower cost. However, their accuracy is restricted by the instability and limited selectivity of the solid-state sensors they rely on [47].

Our goal is to develop a progressive multi-fidelity surrogate model to perform HF estimates of benzene, C_6H_6, concentration (in $\mu g/m^3$) – a key pollutant related to respiratory illness – from a hierarchical set of four measurements provided by an LF multi-sensor. This example serves as a proof of concept for the development of an MF software capable of recovering HF estimates by calibrating LF measurements, thereby broadening accurate monitoring capabilities in resource-constrained settings. The LF multi-sensor

	Level 0 (input param.)	Level 1 (drag/lift)	Level 2 (outflow)	Level 3 (partial domain)
Relative error	12.3%	8.73%	6.82%	6.50%

Table 2: Relative error with respect to the HF reference solution, $\mathbf{y}$, is evaluated for the MF estimates, $\hat{\mathbf{y}}^{(l)}$, at each level l, as $\mathrm{err}_{\%}^{(l)} = \frac{100\%}{N_{\mathrm{test}}} \sum_{n=1}^{N_{\mathrm{test}}} \frac{\left\|\mathbf{y}(t_n,\mu_n) - \hat{\mathbf{y}}^{(l)}(t_n,\mu_n)\right\|_2}{\|\mathbf{y}(t_n,\mu_n)\|_2}$, where N_{test} is the number of time-parameter combinations in the test set.

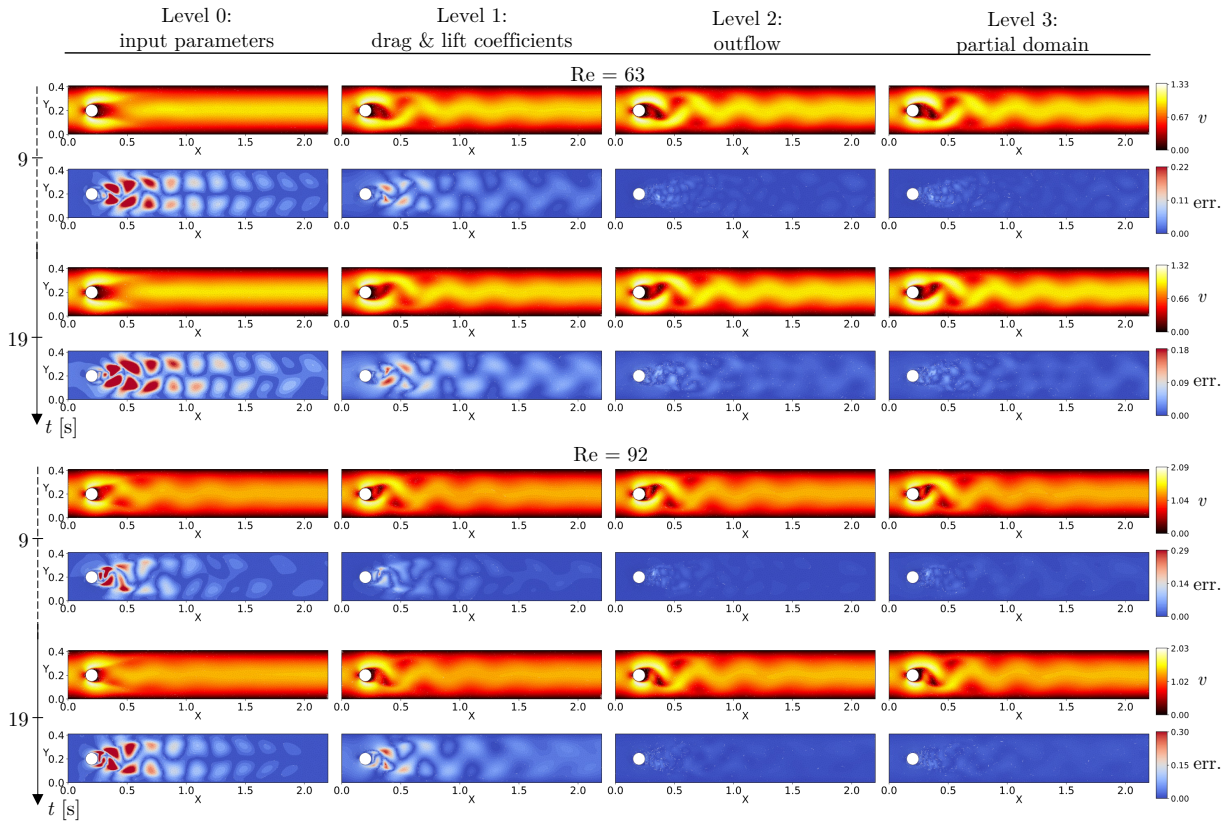

Figure 6: Progressive multi-fidelity prediction of velocity magnitude v for each fidelity level and absolute error with respect to the high-fidelity reference solutions of the Navier-Stokes problem. The illustrated snapshots refer to one interpolated time instance $t = 9$s and one extrapolated time instance $t = 19$s (with $T_{\text{train}} = 15$s marking the end of the HF training coverage) for two testing values of Reynolds number $\mu \in \{63, 92\}$ (top and bottom plots, respectively), unseen during the training.

device [47] is equipped with metal oxide chemoresistive sensors for pollutant measurements, designed to be independently replaceable for ease of maintenance. In addition, it includes low-cost commercial sensors for temperature and relative humidity. The LF measurements are organized into a multi-fidelity hierarchy.

5.1. Multi-fidelity setting

The first two signals, $x^{(0)}$ and $x^{(1)}$, are temperature (in C°) and relative humidity (in %), respectively. These LF signals are inexpensive to obtain and could capture general trends of heating and meteorological influences, but they are poorly correlated with specific pollutant levels. The following two signals, $x^{(2)}$ and $x^{(3)}$, are LF measurements of the concentration (in µg/m^3) of related pollutants, CO and O_3, which do not directly estimate benzene, yet offer valuable complementary information. Importantly, these LF measurements can be acquired at much lower cost than those from HF industrial spectrometers. As shown in Fig. 7, the LF signals $\{x^{(l)}\}_{l=0}^{3}$ are provided as input to the progressive multi-fidelity model to estimate the HF benzene concentration y (in µg/m^3). Outputs at each level are indicated as $\{\hat{y}^{(l)}\}_{l=0}^{3}$ and are expected to progressively improve in accuracy and reliability as more LF information is assimilated while proceeding up the hierarchy.

Dataset. Both LF inputs (at each level) and HF outputs are one-dimensional time series measurements. Therefore all encoder and decoder networks, $\Phi^{(l)}$ and $\Psi^{(l)}$ respectively, are LSTM networks with dimensions $d_{\text{in}}^{(l)} = d_{\text{h}}^{(l)} = d_{\text{out}}^{(l)} = 1$, for $l = 0, \dots, 3$.

Input-output data are available to train the progressive MF model from $T_0 = 2004/10/03$ (YYYY/MM/DD) to $T_{\text{train}} = 2005/01/16$. All measurements are recorded on an hourly basis. Each level is trained $m = 15$

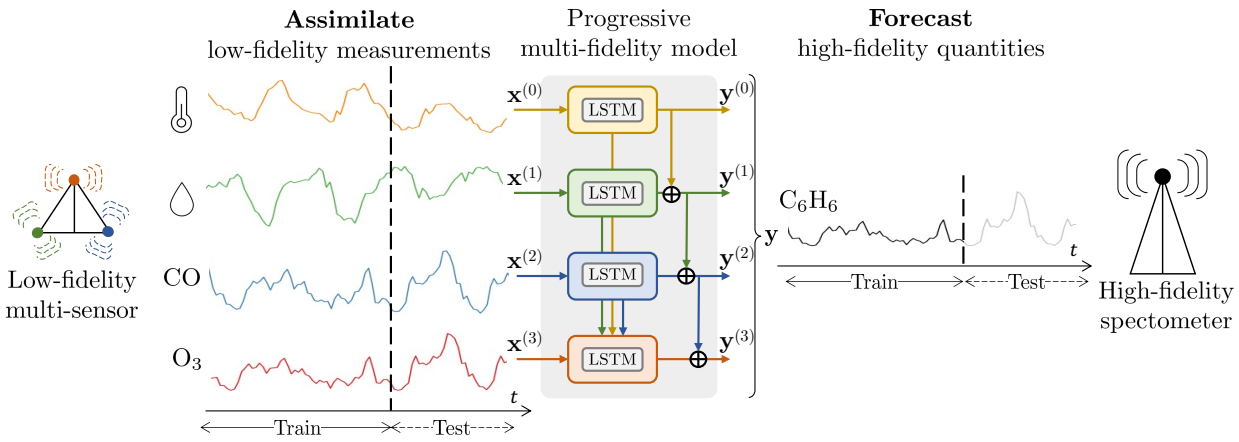

Figure 7: Multi-fidelity set-up for estimating air pollution. A low-fidelity multi-sensor device provides cheap measurements of temperature, humidity and concentrations of CO and O_3. A progressive multi-fidelity model is trained to provide a hierarchy of approximations of the output quantity of interest, which is the high-fidelity estimation of the benzene (C_6H_6) measured by an expensive spectrometer. High-fidelity output data are available for a limited time window for training, while outputs are forecast by the multi-fidelity model at different levels of refinement, according to the availability of low-fidelity measurements.

times to construct an ensemble of models, which are used to provide uncertainty quantification. After T_{train}, we assume to assimilate only LF data up to $T_{\text{test}} = 2005/04/04$ and we test the performance of our model in estimating the HF output in $[T_{\text{train}}, T_{\text{test}}]$.

The data used in this work were provided by De Vito *et al.* [47], collected using a gas multisensor device deployed in an Italian city. The dataset is publicly available at UCI Machine Learning Repository.

5.2. Results

The top plot in Fig. 8 showcases the comparison between the last level prediction of the MF surrogate model and the reference solution, demonstrating high accuracy in reconstructing HF predictions solely from LF measurements. Data are aggregated on a daily basis.

In the bottom plot, two zoomed-in views provide a more detailed analysis of the method's performance across the four levels and at an hourly scale. The left subplot focuses on a period where the model performs particularly well (days $2005/02/15 - 20$), while the one on the right highlights the most significant discrepancies (days $2005/03/9 - 14$). In both scenarios, we observe the progressive enhancement in accuracy as we traverse the hierarchy of the MF model. Each level progressively refines the estimated benzene concentration by integrating the information from the corresponding inputs and all preceding levels.

In particular, at the first levels $l = 0$ and $l = 1$, the model relies on temperature and humidity signals, which offer general environmental context but exhibit limited direct correlation with benzene concentration. Despite these limitations, predictions at these levels effectively capture long-term trends and provide reasonably reliable estimates. However, discrepancies arise in certain periods, where the lack of pollutant-specific data leads to noticeable deviations from the actual HF reference values. Importantly, since temperature and humidity measurements are easy to obtain and consistently available, the progressive MF model ensures continuity in operation. Indeed, even if pollutant sensors ($x^{(2)}$ for CO and $x^{(3)}$ for O_3) require maintenance or break down, the lower-fidelity levels can still provide reasonable estimates, preventing complete model downtime.

	Level 0 (Temperature)	Level 1 (Humidity)	Level 2 (CO)	Level 3 (O_3)
Relative error	72%	66%	29%	13%

Table 3: Relative error with respect to the HF data, y, is evaluated for the MF estimates, $\hat{y}^{(l)}$, at each level l, as $\text{err}_{\%}^{(l)} = \frac{100\%}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \frac{\left\| y(t_n) - \hat{y}^{(l)}(t_n) \right\|_2}{\|y(t_n)\|_2}$, where N_{test} is the number of time instants in the test set.

A substantial improvement is observed at level $l = 2$, where LF measurements of CO concentration are incorporated. This additional input introduces crucial chemical context, significantly enhancing the model's ability to resolve short-term fluctuations in benzene concentration. As a result, level 2 predictions align more closely with the HF ground truth, substantially reducing errors observed at previous levels.
At the final level, $l = 3$, where O_3 concentration is assimilated, the model attains its highest accuracy. The progressive assimilation of information allows this level to make precise corrections, achieving great alignment with HF measurements.
Crucially, even in the most challenging time window (days $2005/03/9 - 14$), the true HF solution is consistently embedded within the model's uncertainty bounds, highlighting the reliability of our approach and its robustness in real-world scenarios where HF data are not available.
Overall, the results validate the effectiveness of our progressive multi-fidelity strategy. The hierarchical structure enables the model to make reasonable predictions even at lower-fidelity levels while progressively refining estimates as richer information becomes available. This adaptability makes our approach particularly suitable for deployment in real-time monitoring systems, where HF data may be limited due to cost or logistical constraints, and ensures operational continuity even in the presence of temporary sensor malfunctions.

6. Conclusions

In this work, we introduced a progressive multi-fidelity learning framework designed to construct surrogate models for physical systems by sequentially integrating heterogeneous, multi-modal datasets. The core of the

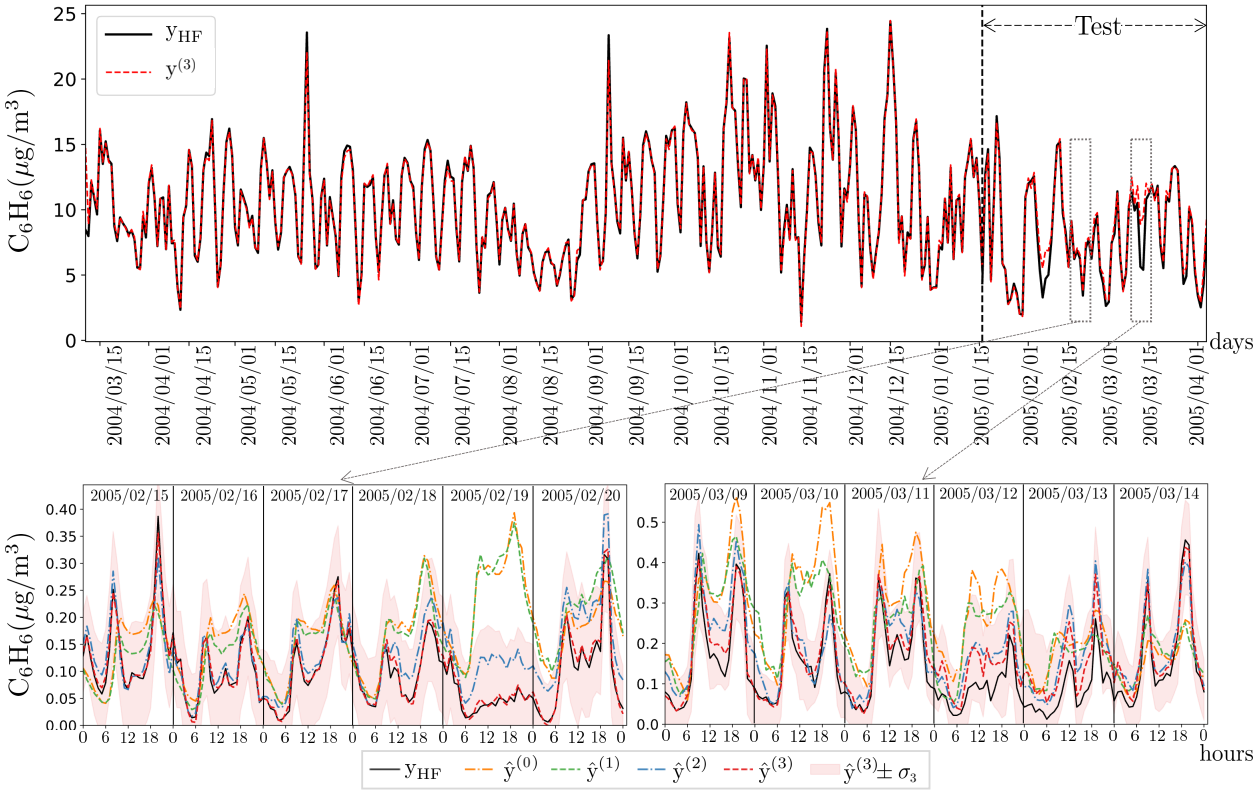

Figure 8: Results for the air pollution test case. The top plot shows the comparison between the actual benzene concentration y_{HF} (black line) and the prediction of the multi-fidelity surrogate at the final level $\hat{y}^{(3)}$ (red dashed line), aggregated daily over the entire time window. The dashed black line at $T_{train} = 2005/01/16$ indicates the end time of HF data coverage in training, i.e. no HF information is available over $t \in [T_{train}, T_{test}]$. The bottom plot provides two zoomed-in views, highlighting the forecasting capabilities of the multi-fidelity surrogates at hourly frequency across all hierarchy levels. The shaded region indicates $\pm$ one standard deviation σ_3 around the prediction at the final level, offering uncertainty quantification for the multi-fidelity estimates.

approach is a hierarchical architecture that processes data at each fidelity level through tailored encoders, fuses the resulting latent representations, and applies additive corrections to predictions from preceding levels. This design ensures that knowledge is retained and refined as new data becomes available, effectively mitigating catastrophic forgetting.

We demonstrated the flexibility and effectiveness of our method across three distinct applications: two involving simulated data (a nonlinear reaction-diffusion system and a Navier-Stokes fluid flow problem) and one real-world task (air pollution monitoring). The results consistently demonstrated that the progressive fusion of information allows the model to efficiently leverage knowledge from low-fidelity data, leading to a systematic increase in prediction accuracy and a corresponding reduction in predictive uncertainty at each level. Crucially, the framework maintains operational continuity, providing reasonable estimates even when only lower-fidelity inputs are available due to computational, budget, or data assimilation constraints. The integrated uncertainty quantification further enhances the model's reliability, even in extrapolative regimes.

Future work will focus on enhancing the framework's flexibility and computational efficiency. A primary goal is to develop a more robust architecture that relaxes the current requirement for a strict data hierarchy, enabling the model to handle arbitrary combinations of available or missing data modalities during inference. To improve efficiency, the computationally expensive ensemble-based uncertainty quantification could be replaced with inherent probabilistic architectures, such as Bayesian neural networks or conditional neural processes [39, 34]. Finally, the generality and effectiveness of the paradigm suggest its applicability to a broader range of scientific computing problems, including inverse problems where the multi-fidelity structure could be naturally embedded within multi-level inference schemes.

Acknowledgment

PC has been supported under the JRC STEAM STM-Politecnico di Milano agreement and by the PRIN 2022 Project "Numerical approximation of uncertainty quantification problems for PDEs by multi-fidelity methods (UQ-FLY)" (No. 202222PACR), funded by the European Union - NextGenerationEU.

AM acknowledges the project "Dipartimento di Eccellenza" 2023-2027 funded by MUR, the project FAIR (Future Artificial Intelligence Research), funded by the NextGenerationEU program within the PNRR-PE-AI scheme (M4C2, Investment 1.3, Line on Artificial Intelligence) and the Project "Reduced Order Modeling and Deep Learning for the real- time approximation of PDEs (DREAM)" (Starting Grant No. FIS00003154), funded by the Italian Science Fund (FIS) - Ministero dell'Università e della Ricerca.

AF acknowledges the PRIN 2022 Project "DIMIN- DIgital twins of nonlinear MIcrostructures with iN-novative model-order-reduction strategies" (No. 2022XATLT2) funded by the European Union - NextGen-erationEU.

PC thanks Johannes Lips for fruitful discussions on the air pollution example.

Appendix A. Appendix

This appendix provides a summary of the key hyperparameters and network architectures used for the different case studies presented in the main text. The specific choices for each application are detailed in Table A.4.

The different *encoder types* reflect the input data modalities of the physical systems, as explained in the *Multi-fidelity setting* Sections 3.1-4.1-5.1. For the encoder neural networks (NNs), the input and output dimensions are given by *input dimension* d_{in} and *latent dimension* d_h, respectively. When a dimension is reported with N_{POD}, the data are first reduced using POD to the specified number of modes, and the network is trained on the resulting POD coefficients. The number of hidden layers and nodes for each NN is indicated as a list, where the number of elements corresponds to the number of hidden layers and the values specify the number of nodes per hidden layer.

For the decoders, the input dimension is given by $d^{(l)}h_{\text{tot}} = \sum_{j\leq l} d_h^{(j)}$ at level l, while the *output dimension* is d_{out} at every level. The decoder architectures and the number of layers/nodes are indicated under *decoder type* and *decoder NN layers/nodes*, respectively, and are the same at every level since a single high-fidelity quantity is modeled.

Moreover, Table A.4 lists general NN hyperparameters used for training. The Adam optimizer was used in all cases, with learning rates and L_2 regularization coefficients tuned individually. The tanh activation

function provided stable training and good performance across all applications. The regularization weights λ_Φ and λ_Ψ for the encoder and decoder networks, respectively, were kept equal to 1.0 in all experiments, with the overall regularization strength λ_{reg} controlled by the reported value.
Finally, m indicates the number of independent retrainings with randomly initialized weights used to compute statistical moments for ensemble-based uncertainty quantification.

References

[1] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting

Hyperparameter	**Reaction–Diffusion**	**Navier–Stokes**	**Air–Pollution**
Encoder type			
Level 0	Dense	Dense	LSTM
Level 1	LSTM	LSTM	LSTM
Level 2	POD+LSTM	LSTM	LSTM
Level 3		AE-LSTM	LSTM
Input dimensions d_{in}			
Level 0: $d_{\text{in}}^{(0)}$	2	2	1
Level 1: $d_{\text{in}}^{(1)}$	4	2	1
Level 2: $d_{\text{in}}^{(2)}$	1024 ($N_{\text{POD}} = 9$)	20	1
Level 3: $d_{\text{in}}^{(3)}$		577	1
Latent dimensions d_h			
Level 0: $d_h^{(0)}$	2	2	1
Level 1: $d_h^{(1)}$	2	2	1
Level 2: $d_h^{(2)}$	4	2	1
Level 3: $d_h^{(3)}$		2	1
Encoder NN layers/nodes			
Level 0	[31]	[20]	[20,20]
Level 1	[31]	[20]	[20,20]
Level 2	[31]	[20]	[20,20]
Level 3		[20]	[20,20]
Output dimensions d_{out}			
same for all levels	10000 ($N_{\text{POD}} = 11$)	8239 ($N_{\text{POD}} = 32$)	1
Decoder type			
same for all levels	LSTM+POD	LSTM+POD	LSTM
Decoder NN layers/nodes			
same for all levels	[25, 25, 25]	[20, 20]	[20,20]
General NN hyperparameters			
Optimizer	Adam	Adam	Adam
Learning rate	$2.8 \cdot 10^{-5}$	10^{-3}	10^{-3}
Activation function	tanh	tanh	tanh
L_2 regularization λ_{reg} (see Eq. (4))	$2.8 \cdot 10^{-6}$	10^{-5}	10^{-6}
λ_Φ (see Eq. (5))	1.0	1.0	1.0
λ_Ψ (see Eq. (5))	1.0	1.0	1.0
#Trainings for ensemble UQ m	30	3	15

Table A.4: Summary of encoder and decoder network architectures and training hyperparameters for the different applications.

with machine learning. *Nature*, 637(8044):84–90, 2025.

[2] Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A Weyn, Haiyu Dong, et al. A foundation model for the earth system. *Nature*, pages 1–8, 2025.

[3] Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P Bruinsma, Tom R Andersson, Michael Herzog, Nicholas D Lane, Matthew Chantry, J Scott Hosking, et al. End-to-end data-driven weather prediction. *Nature*, 641(8065):1172–1179, 2025.

[4] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

[5] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction*. Springer International Publishing, 2016.

[6] Peter Benner, Volker Mehrmann, and Danny C Sorensen. *Dimension reduction of large-scale systems*, volume 45. Springer, 2005.

[7] Bernd R Noack, Marek Morzynski, and Gilead Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.

[8] Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer International Publishing, 2016.

[9] Kookjin Lee and Kevin T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[10] Stefania Fresca, Luca Dede, and Andrea Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized pdes. *Journal of Scientific Computing*, 87(2):1–36, 2021.

[11] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *SIAM Review*, 60(3):550–591, 2018.

[12] Paolo Conti, Mengwu Guo, Andrea Manzoni, Attilio Frangi, Steven L Brunton, and J Nathan Kutz. Multi-fidelity reduced-order surrogate modelling. *Proceedings of the Royal Society A*, 480(2283):20230655, 2024.

[13] Shady E Ahmed, Omer San, Kursat Kara, Rami Younis, and Adil Rasheed. Multifidelity computing for coupling full and reduced order models. *Plos one*, 16(2):e0246092, 2021.

[14] Mariella Kast, Mengwu Guo, and Jan S Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Comput. Methods Appl. Mech. Engrg.*, 364:112947, 2020.

[15] Nicholas Geneva and Nicholas Zabaras. Multi-fidelity generative deep learning turbulent flows. *Foundations of Data Science*, 2(4):391–428, 2020.

[16] Anthony O'Hagan and Marc C. Kennedy. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[17] Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2198):20160751, 2017.

[18] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012.

[19] Rémi Lam, Douglas L Allaire, and Karen E Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143, 2015.

[20] Mengwu Guo, Andrea Manzoni, Maurice Amendt, Paolo Conti, and Jan S Hesthaven. Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities. *Computer methods in Applied Mechanics and Engineering*, 389:114378, 2022.

[21] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.

[22] Mohammad Motamed. A multi-fidelity neural network surrogate sampling method for uncertainty quantification. *International Journal for Uncertainty Quantification*, 10(4), 2020.

[23] Dehao Liu and Yan Wang. Multi-fidelity physics-constrained neural network and its application in materials modeling. *Journal of Mechanical Design*, 141(12), 2019.

[24] Paolo Conti, Mengwu Guo, Andrea Manzoni, and Jan S Hesthaven. Multi-fidelity surrogate modeling using long short-term memory networks. *Computer methods in applied mechanics and engineering*, 404:115811, 2023.

[25] Sandra Steyaert, Marija Pizurica, Divya Nagaraj, Priya Khandelwal, Tina Hernandez-Boussard, Andrew J Gentles, and Olivier Gevaert. Multimodal data fusion for cancer biomarker discovery with deep learning. *Nature machine intelligence*, 5(4):351–362, 2023.

[26] Kevin M Boehm, Pegah Khosravi, Rami Vanguri, Jianjiong Gao, and Sohrab P Shah. Harnessing multimodal data integration to advance precision oncology. *Nature Reviews Cancer*, 22(2):114–126, 2022.

[27] Yu-Dong Zhang, Zhengchao Dong, Shui-Hua Wang, Xiang Yu, Xujing Yao, Qinghua Zhou, Hua Hu, Min Li, Carmen Jiménez-Mesa, Javier Ramirez, et al. Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. *Information Fusion*, 64:149–187, 2020.

[28] Girish Narayanswamy, Xin Liu, Kumar Ayush, Yuzhe Yang, Xuhai Xu, Shun Liao, Jake Garrison, Shyam Tailor, Jake Sunshine, Yun Liu, et al. Scaling wearable foundation models. *arXiv preprint arXiv:2410.13638*, 2024.

[29] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[30] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[32] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[33] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[34] Dongxia Wu, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Multi-fidelity hierarchical neural processes. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2029–2038, 2022.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, abs/1412.6980, 2014.

[36] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[37] David JC MacKay. Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469, 1995.

[38] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[39] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR, 2018.

[40] Johannes Lips, Hendrik Lens, and Paolo Conti. Soft sensor design using hierarchical multi-fidelity modeling with bayesian optimization for input variable selection. *IFAC-PapersOnLine*, 59(6):139–144, 2025.

[41] Lloyd N Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.

[42] B.N. Rajani, A. Kandasamy, and Sekhar Majumdar. Numerical simulation of laminar flow past a circular cylinder. *Appl. Math. Mod.*, 33(3):1228 – 1247, 2009.

[43] Momchilo M Zdravkovich. *Flow around circular cylinders: Volume 2: Applications*, volume 2. Oxford university press, 1997.

[44] Michael Schäfer, Stefan Turek, Franz Durst, Egon Krause, and Rolf Rannacher. Benchmark computations of laminar flow around a cylinder. In *Flow simulation with high-performance computers II: DFG priority research programme results 1993–1995*, pages 547–566. Springer, 1996.

[45] Federico Negri. redbKIT Version 2.2. `http://redbkit.github.io/redbKIT/`, 2016.

[46] David Mage, Guntis Ozolins, Peter Peterson, Anthony Webster, Rudi Orthofer, Veerle Vandeweerd, and Michael Gwynne. Urban air pollution in megacities of the world. *Atmospheric environment*, 30(5):681–686, 1996.

[47] Saverio De Vito, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008.