FUSE: Fast Unified Simulation and Estimation for PDEs

Levi E. Lingsch
Seminar for Applied Mathematics
& AI Center, ETH Zurich
levi.lingsch@ai.ethz.ch

Dana Grund
Institute for Atmospheric
and Climate Science, ETH Zurich
dana.grund@ethz.ch

Siddhartha Mishra
Seminar for Applied Mathematics
& AI Center, ETH Zurich
siddhartha.mishra@ethz.ch

Georgios Kissas
AI Center
ETH Zurich
gkissas@ai.ethz.ch

Abstract

The joint prediction of continuous fields and statistical estimation of the underlying discrete parameters is a common problem for many physical systems governed by PDEs. Hitherto, it has been separately addressed by employing operator learning surrogates for field prediction while using simulation-based inference (and its variants) for statistical parameter determination. Here, we argue that solving both problems within the same framework can lead to consistent gains in accuracy and robustness. To this end, we propose a novel and flexible formulation of the operator learning problem that allows jointly predicting infinite-dimensional quantities and inferring distributions of finite-dimensional parameters and thus amortizing the cost of both the inverse and the surrogate models to a joint training step. We present the capabilities of the proposed methodology for predicting continuous and discrete biomarkers in full-body haemodynamics simulations under different levels of missing information, and in a test case for atmospheric large-eddy simulation of a two-dimensional dry cold bubble, predicting continuous time-series measurements from inferred system conditions. For both cases, we present comparisons against different baselines to demonstrate significantly increased accuracy in both the inverse and surrogate tasks.

1 Introduction

Partial Differential Equations (PDEs) describe the propagation of system conditions for a very wide range of physical systems. Parametric PDEs consider different system conditions as well as an underlying solution operator characterized by a set of finite-dimensional parameters. Traditional numerical methods based on different discretization schemes such as Finite Differences, Finite Volumes, and Finite Elements have been developed along with fast and parallelizable implementations to tackle complex problems, such as atmospheric modeling and cardiovascular biomechanics. For parametric PDEs, these methods define maps from the underlying set of parameters, which describe the dynamics and the boundary/initial conditions and which we assume to be finite-dimensional here, to physical quantities such as velocity or pressure that are continuous in the spatio-temporal domain. Despite their successful application, there still exist well-known drawbacks of traditional solvers. To describe a particular physical phenomenon, PDE parameters and solvers need to be calibrated on precise conditions that are not known a priori and cannot easily be measured in realistic applications. Therefore, iterative and thus expensive calibration procedures are considered in the cases where the parameters and conditions are inferred from data [2]. Even after the solvers are calibrated, an

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

ensemble of solutions needs to be generated to account for uncertainties in the model parameters or assess the sensitivity of the solution to different parameters, which are computationally prohibitive downstream tasks [43].

Related work A variety of deep learning algorithms have recently been proposed for scientific application to PDEs, broadly categorized into surrogate and inverse modeling algorithms, to either reduce the computational time of complex forward simulations or infer missing finite-dimensional information from data to calibrate a simulator to precise conditions.

Surrogate learning is a paradigm for accelerating computations. Hence, the cost of evaluating continuous quantities is amortized to an offline training stage. For functional data, the so-called *Neural Operators* [28, 3] generalizes to different conditions and discretizations of a complex system (resolution invariance). A multitude of operator learning approaches has been designed [37, 27, 50, 21, 44, 33, 32, 17], such as the Fourier Neural Operator (FNO) [31]. However, a priori, neural operators are not designed to process maps between finite-dimensional and continuous quantities. A different surrogate modeling strategy is constructed by learning a map between PDE parameters and solutions, as in traditional reduced-order model (ROM) approaches [16, 8, 19], and their deep learning counterparts [23, 10], including Generative Adversarial ROM (GAROM) [10]. In addition, Gaussian processes and kriging have been revisited in the context of PDE emulation [15, 61].

Amortizing the cost of *parameter inference* has also been widely explored in the literature of generative modeling and Simulation-Based Inference (SBI). In SBI, Invertible Neural Networks are combined with discrete Normalizing Flows to develop Neural Posterior Estimation [11, 12, 24, 45] or continuous Normalizing Flows in Flow Matching Posterior Estimation (FMPE) [60, 35]. However, they commonly rely on an expensive physical solver to sample continuous predictions, and are hindered by the use of approximate physical models, Gaussian distributions, or a limited sample size.

Approaches that attempt to *jointly infer parameters and learn a surrogate* are much rarer in the literature. Many are built on Variational Autoencoders (VAEs) [57], in finite [25] or infinite [51] dimensions. They inherently assume an underlying bijective relation between the functional and scalar quantities or are restricted to Gaussian latent representations, which limits their applicability to nonlinear problems. Among them, InVAErt [57] extends a deterministic encoder-decoder pair with a variational latent encoder. However, InVAErt is primarily constructed for investigating identifiability of parameters in systems of PDEs, and may not be suitable for learning uncertainty propagation from parameters to continuous fields. The conditional Variational Neural Operator (cVANO) [26] makes use of the resolution invariance of neural operators, based on [51]. cVANO relies on a VAE to learn a low-dimensional manifold of the system constrained by finite-dimensional parameters, but is restricted to Gaussian approximations which are not suitable for problems which lack bijectivity. As alternative approaches, Simformer [14] estimates both finite-dimensional and function-valued parameters using transformers within the SBI framework, and PAGP [62] leverages the expressive power of Gaussian Processes. However, both are not suited for functional input due to quadratic growth with the input size. Finally, OpFlow [53] brings together operator learning and normalizing flows for quantifying uncertainty in output functions, but it lacks interpretability and inference on the latent space.

Our contribution Based on the experience and limitations of the aforementioned approaches, we formulate the following requirements for a unified forward-inverse framework relating functional quantities to a known space of finite-dimensional parameters: First, the interpretability provided by the given parameterization should be retained in both the forward and inverse tasks as well as their combination. Then, both the forward and inverse model should be discretization invariant with respect to all functional variables. Finally, a general probabilistic representation should be chosen for the latent parameter space, allowing for arbitrary parameter distributions. Following these guidelines, the main contributions of our paper are the following.

- We propose *Fast Unified Simulation and Estimation for PDEs* (FUSE), a rigorous framework for unified inverse and forward problems for parametric PDEs. The FUSE framework, illustrated in Figure 1, allows to combine different forward and inverse models, and we choose to instantiate it with Fourier Neural Operators (FNOs) and Flow-Matching Posterior Estimation (FMPE) for the experiments in this work.

- We formulate a mathematical framework with a *unified objective* for the forward and inverse problems, which allows us to assess how uncertainty in the inverse problem propagates

2

through the forward problem (*propagated uncertainty*) and evaluate both models together to avoid nonlinear error amplification at model concatenation.

- We showcase how to *adapt forward surrogate and inverse estimation models* to fit the FUSE framework, mapping between finite and infinite-dimensional spaces. In particular, we extend FNO with a custom lifting to take finite-dimensional inputs, and we equip FMPE with an FNO-based encoding to allow for functional conditional information.

- Our *experiments* show that our implementation of FUSE overcomes struggles of baselines (cVANO, GAROM, InVAErt) and an ablation (U-Net) on two complex and realistic PDE examples, pulse wave propagation (PWP) in the human arterial network and an atmospheric cold bubble (ACB). The experiments exemplify its ability for accurate and fast surrogate modeling, parameter inference, out-of-distribution generalization, and the flexibility to handle different levels of input information.

To our knowledge, the extensions we present to FNO and FMPE have not been explored in the literature yet. We would like to emphasize that we chose to base ourselves on these methods to illustrate the FUSE framework, but other forward and inverse methods may be more suitable for particular test cases.

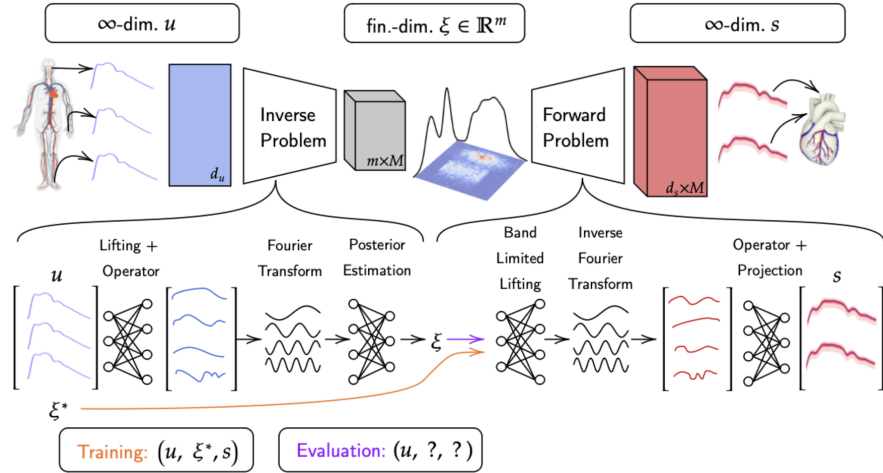



Figure 1: FUSE models a posterior distribution over finite-dimensional parameters ξ given infinite-dimensional functions u with d_u components (channels). It learns other continuous functions s with d_s channels from the parameters ξ. Band-limited Fourier transforms and a lifting operator act as a bridge between finite and infinite dimensions for the forward problem. Likewise, as inference models such as FMPE or NPE require fixed-size inputs, the inverse operator layers are conjoined with a band-limited Fourier transform to learn a fixed-size representation of the input function.

2   Methods

Notation and Assumptions   Let $\mathcal{U} \subset \mathcal{C}(X, \mathbb{R}^{d_u})$ and $\mathcal{S} \subset \mathcal{C}(Y, \mathbb{R}^{d_s})$ be spaces of continuous functions on compact domains $X \subseteq \mathbb{R}^d$ and $Y \subseteq \mathbb{R}^{d'}$, respectively, and let $\Xi \subseteq \mathbb{R}^m$ be a space of finite-dimensional parameters. For a map $F : \mathcal{A} \to \mathcal{B}$, where we take $\mathcal{A}$ and $\mathcal{B}$ to be among the spaces $\mathcal{U}$, $\mathcal{S}$, or Ξ, and a probability measure $\pi \in \text{Prob}(\mathcal{A})$, we denote by $F_{\#\pi} \in \text{Prob}(\mathcal{B})$ the push-forward measure that expresses uncertainty on a set $B \subseteq \mathcal{B}$ by the propagation of π through F, defined as $F_{\#\pi}(B) = \pi(F^{-1}(B))$. Further, we assume all metrics on measures in this paper to be the total variation metric, and denote them by d both on $\text{Prob}(\mathcal{S})$ and $\text{Prob}(\Xi)$. In general, we assume all maps to be Lipschitz continuous, and we omit Lipschitz constants in inequalities (details are provided in the Appendix).

Problem Formulation   Consider the setting of a parametric PDE with forward solution mapping $\mathcal{G} : \xi \mapsto s$, where $\xi \in \Xi$ is a vector of finite-dimensional parameters and $s \in \mathcal{S}$ is a function-valued output. Since it is the goal to calibrate the input parameters, we omit any functional inputs to $\mathcal{G}$, such

3

as initial or boundary conditions, and assume they are kept constant or encoded in the parameters ξ. Given the forward operator $\mathcal{G}$, the *forward model uncertainty* is quantified by propagating a given distribution of parameters $\rho \in \mathrm{Prob}(\Xi)$ through the model, i.e. evaluating the push-forward measure $\mathcal{G}_{\#\rho}$. In practice, the parameters ξ are not available and need to be inferred from indirect measurements $u \in \mathcal{U}$, which we assume to be functions in space or time (e.g., time series), and, in general, $u \neq s$. The extended solution operator $\tilde{\mathcal{G}} : u \mapsto s$ then maps between functional measurements and functional model output, omitting the intermediate parameter space. Since the measurements u are impacted by the uncertain parameters ξ, the function inputs u are as well equipped with uncertainty and represented by the measure $\mu \in \mathrm{Prob}(\mathcal{U})$. Finally, the inverse problem consists in estimating the distribution $\rho(\xi|u)$ of parameters given measurements u. The corresponding uncertainty on the predicted outcomes s is then given by the *propagated uncertainty* $\mathcal{G}_{\#\rho(.|u)}$.

Unified Objective   Given an approximate forward operator $\tilde{\mathcal{G}}^{\theta} \approx \tilde{\mathcal{G}}$ between function spaces, and an estimated distribution $\mu^{\phi} \approx \mu$ of input functions u, parameterized by θ and ϕ, respectively, we use triangle inequality to observe that

$$d(\tilde{\mathcal{G}}^{\theta}_{\#\mu^{\phi}}, \tilde{\mathcal{G}}_{\#\mu}) \leq \underbrace{d(\tilde{\mathcal{G}}^{\theta}_{\#\mu^{\phi}}, \tilde{\mathcal{G}}^{\theta}_{\#\mu})}_{\text{Measure matching}} + \underbrace{d(\tilde{\mathcal{G}}^{\theta}_{\#\mu}, \tilde{\mathcal{G}}_{\#\mu})}_{\text{Operator learning}}. \tag{1}$$

Thus, we found a unified objective consisting of two steps corresponding to the two terms in the right hand side (rhs) of Equation (1). In the measure matching step, the objective amounts to learn an approximation μ^{ϕ} of the measure μ, whereas in the operator learning step, the objective is to learn a Neural Operator $\tilde{\mathcal{G}}^{\theta}$ that approximates the underlying ground truth operator $\tilde{\mathcal{G}}$.

Our reformulation of operator learning naturally fits into the aim of this paper to propose a method that can act as an operator surrogate (*forward problem*, operator learning objective) as well as performing parameter inference by minimizing distances on measures (*inverse problem*, measure matching objective), unifying the apparently unrelated problems of surrogate modeling and inference.

Forward Problem: Operator Learning   The operator $\tilde{\mathcal{G}}$ resembles the solution operator to a PDE that maps function inputs such as initial and boundary conditions, coefficients, sources etc to (observables of) the solution of the PDE. It is the goal of supervised operator learning [29] to learn this type of operator as a parameterized Neural Operator (NO) $\tilde{\mathcal{G}}^{\theta}$ on a (training) distribution $\mu \in \mathrm{Prob}(\mathcal{U})$ by minimizing the operator learning objective in (1). This objective is bound by *supervised learning* on the parameter space Ξ,

$$d(\tilde{\mathcal{G}}^{\theta}_{\#\mu}, \tilde{\mathcal{G}}_{\#\mu}) \leq d(\mathcal{G}^{\theta}_{\#\rho}, \mathcal{G}_{\#\rho}), \tag{2}$$

and further by the term (Appendix A.6),

$$\mathcal{L}_1(\theta) = \int_{\Xi} \|\mathcal{G}^{\theta}(\xi) - \mathcal{G}(\xi)\|_{L^1} d\rho(\xi). \tag{3}$$

The loss is approximated by training samples of the form $\{\xi^i, \mathcal{G}(\xi^i)\}_{i=1}^{N}$, sampled from an underlying data distribution $\rho \in \mathrm{Prob}(\Xi)$. In order to use a NO on finite-dimensional inputs, we define the *band-limited lifting* $h^{\theta_3}(\xi)$, composed of a lifting increasing the dimension of the parameters and an inverse Fourier transform. We then choose to instantiate $\mathcal{G}^{\theta}$ based on Fourier Neural Operator (FNO) [31] layers $\mathcal{K}^{\theta_2}$, implemented with discrete spectral evaluations as in [34] to handle irregularly sampled measurements for the output function,

$$\mathcal{G}^{\theta}(\xi) = \mathcal{Q}^{\theta_1} \circ \mathcal{K}^{\theta_2} \circ h^{\theta_3}(\xi), \tag{4}$$

where $\mathcal{Q}^{\theta_1}$ is a learnable map that projects the channels to the dimensions of the output function and $\theta = [\theta_1, \theta_2, \theta_3]$ are the trainable parameters.

Inverse Problem: Measure Matching   We would like to approximate the true posterior measure $\rho(\xi|u)$ given measurements u by $\rho^{\phi}(\xi|u)$. This task is equivalent to minimizing the measure matching objective in (1) since (Appendix A.7)

$$d(\tilde{\mathcal{G}}^{\theta}_{\#\mu^{\phi}}, \tilde{\mathcal{G}}^{\theta}_{\#\mu}) \leq d(\rho^{\phi}(\xi|u), \rho(\xi|u)). \tag{5}$$

To minimize the latter distance, we adapt Flow-Matching Posterior Estimation (FMPE) to handle function-valued conditional inputs u. FMPE trains a flow function that maps samples from a standard

4

normal distribution to those from the target distribution $\rho(\xi|u)$. Unlike diffusion models [54, 20, 58], which require multiple steps, FMPE achieves this transformation in a single step, enabling faster training and evaluation. Both approaches are similar, but differ in their probability paths: diffusion models use a diffusion-based path, while FMPE employs an optimal transport path, leading to greater regularity and improved performance in low data settings.

The function input u is handed to the FMPE flow parameterized by $v_{t,\hat{u}}^{\phi_0}(\xi)$ as finite-dimensional conditional information $\hat{u}$. We encode the infinite-dimensional u into $\hat{u} = T^{\phi_1,\phi_2}(u)$ as an adapted FNO, mimicking the forward surrogate. We project the functions to the parameters using a band-limited forward Fourier transform $\tilde{h}$ to create a fixed size latent representation of a function regardless of the input dimensions, subsequently projecting the large latent dimension to the smaller dimension m of ξ by FMPE,

$$\hat{u} = T^{\phi_1,\phi_2}(u) = \tilde{h} \circ \mathcal{K}^{\phi_1} \circ \mathcal{P}^{\phi_2}(u), \tag{6}$$

where $\mathcal{P}^{\phi_2}$ is a lifting layer and $\mathcal{K}^{\phi_1}$ a concatenation of FNO layers. For the detailed formulation of the FMPE loss, we refer to the original reference [60] and equation (32) in Appendix A.8, and formulate it simply as

$$\mathcal{L}_2(\phi) = \mathcal{L}^{FMPE}(\phi), \tag{7}$$

where $\phi = [\phi_0, \phi_1, \phi_2]$ are the joint trainable parameters of the flow itself and the FNO-based conditional information. The flow is then trained on data of the form $\{(u_i, \xi_i)\}_{i=1}^n$.

Unified Training and Evaluation   The proposed methodology FUSE obtains the optimally trained parameters θ and ϕ by minimizing the respective objectives of Equations (3) and (7),

$$\mathcal{L}_{\text{FUSE}}(\theta, \phi) = \mathcal{L}_{\text{FUSE}}^{\text{forward}}(\theta) + \mathcal{L}_{\text{FUSE}}^{\text{inverse}}(\phi) = \mathcal{L}_1(\theta) + \mathcal{L}_2(\phi), \tag{8}$$

thus minimizing the joint objective (1). For all experiments presented here, the two FUSE objectives are decoupled during training, implying that for each training batch, $\mathcal{L}_1$ and $\mathcal{L}_2$ are calculated and backpropagation is performed separately over each set of network parameters. This strategy aids the training, allowing us to avoid hyperparameter tuning of the loss function. It is possible to backpropagate losses through the entire network as discussed in Appendix A.1; however, this does not result in improvements at evaluation time, further supporting our claim that decoupled objectives may be unified at evaluation time to draw connections between forward and inverse problems.

When employing the trained FUSE model, we assume that measurements u are available and it is the goal to infer both the distribution of parameters ξ that best fit the data and the distribution of function outputs s that would result from those parameters. We obtain M parameter samples from the trained FMPE model as $\xi_i \sim \rho^\phi(\xi|u)$, $i = 1, ..., M$, and pass these to the FNO forward surrogate to obtain M samples of the simulated outputs $s_i = \mathcal{G}^\theta(\xi_i)$, $i = 1, ..., M$, where we assume M sufficiently large to adequately sample the parameter distribution. We hence obtain an ensemble of estimates with mean prediction $\bar{s} = \frac{1}{M} \sum_{i=1}^M s_i$ and standard deviation $\sigma_s = \left(\frac{1}{M-1} \sum_{i=1}^M (s_i - \bar{s})^2\right)^{1/2}$. While both model parts can also be evaluated separately, it is this *propagated uncertainty* on s that offers most insight in the effect of parametric uncertainty within the emulated models.

3   Experiments

For evaluation, we are given testing data of the form (u, ξ^*, s) obtained by numerical simulation, where we will refer to ξ^* as the *data-generating parameters*. Based on measurements u, the inverse FMPE model provides samples of the distribution of discrete parameters ξ_i, that are compared to the true data-generating parameters ξ^* using the Continuous Ranked Probability Score (CRPS). The performance of the deterministic forward prediction is measured with the relative L^1 and L^2 errors over the continuous data and predictions. We compare FUSE against cVANO [26], inVAErt [57], and GAROM [10], and we consider an ablation where we substitute the FNO layers with UNet layers [47] in the FUSE model and leave out the Fourier transforms. In addition, we provide an ablation using a denoising diffusion probabilistic model (DDPM) [20] instead of FMPE in Table 5, and examine FMPE's performance with varying data sizes in Table 6. For details on the training and evaluation metrics, please refer to A.2.

PDE Test Cases As test cases, we consider a simulation of the pulse wave propagation (PWP) in the human arterial network and one of an atmospheric cold bubble (ACB). Both experiments are

5

described by a set of PDEs whose solutions are fully parameterized by finite-dimensional parameters encoding model properties, as well as boundary and initial conditions. In both of these experiments, to replicate a patient or downburst observed in the real world, the solver needs to be calibrated for precise conditions, e.g. a specific patient or atmospheric conditions, and forward uncertainty quantification of the calibrated predictions are of interest. Details on both cases are provided in Appendices A.3 and A.4.

Atmospheric Cold Bubble (ACB) In atmospheric modeling, the two-dimensional dry cold bubble [56] is a well-known test case for numerical simulations resembling downbursts, which cause extreme surface winds in thunderstorms. Within a domain of neutral atmospheric stability without background wind, an elliptic cold air anomaly is prescribed and initiates a turbulent flow as it sinks to the ground, see Figure A.12 for an example. Time series of horizontal and vertical velocity, u and w, are recorded at eight sensors placed inside the domain. Similar data is obtained by weather stations ($z \approx 2$ m) or turbulence flux towers ($z \approx 10 - 50$ m), experienced by wind turbines or high-rise buildings ($z \approx 100$ m), or with unmanned aerial vehicles. The PDE model PyCLES [42] used for simulation is parameterized by the turbulent eddy viscosity ν_t and diffusivity D_t as model parameters, as well as four parameters describing the amplitude and shape of the initial cold perturbation (a, x_r, z_r, z_c). The target task for FUSE here is to calibrate these parameters $\xi \in \mathbb{R}^6$ to time series measurements $u : [0, T] \to \mathbb{R}^{20}$, at ten locations for horizontal and vertical velocity each, where the input measurements used for calibration and the model output used for prediction and uncertainty quantification share the same space $\mathcal{U} = \mathcal{S}$ of time series.

Pulse Wave Propagation (PWP) in the Human Body The pulse wave propagation in the cardio-vascular system contains a great deal of information regarding the health of an individual. For this reason, there are efforts to measure and leverage PWP in both wearable devices, e.g. smart-watches, and clinical medicine. Different systemic parameters $\xi \in \mathbb{R}^{32}$ such as stroke volume (SV), heart rate (HR), and patient age have been shown to affect the morphology of pulse waves [7]. These are used to parameterize pulse wave time series of pressure, velocity, and photoplethysmography (PPG) at 13 locations of systemic arteries of the human cardiovascular system through a reduced order PDE model in the dataset published by [7]. See Tables 7 and 8 in Section A.3 for the full set of parameters and locations. In clinical applications, only measurements u at easily accessible locations (such as the wrist) are available, while it is the goal to predict the pulse signal s at other locations of interest from the inferred parameters ξ. The space of input functions $\mathcal{U}$ hence differs from the space of output functions $\mathcal{S}$ in this test case. In order to account for different clinical scenarios, we train the FUSE model using random masking of locations and evaluate it on different levels of available input information,

Level 1. **Perfect information:** Pressure, velocity, and PPG at all locations,

Level 2. **Intensive care unit information:** Pressure, velocity, and PPG at the wrist,

Level 3. **Minimal information:** PPG at the fingertip.

4   Results

Collected results for all experiments are summarized in Table 1. While the main tasks of the FUSE model are evaluated for both test cases, we focus on ACB to exemplify its generalization properties and on PWP for different levels of input information. We would like to emphasize that the goal of uncertainty representation here is to capture parametric uncertainty inherent to the data-generating numerical model and its parameterizations. Our model does not provide a notion of uncertainty in the predictions due to imperfect training of the neural networks. Given that these errors are generally low (Table 1), we are confident to interpret all spread on ξ and s given by FUSE as parametric uncertainty in the data-generating model. In particular, we expect the true posterior distributions to have positive spread, hence a prediction with small ensemble spread is not necessarily accurate.

Inverse Problem Given time series measurements u, the inverse model samples from the distribution of parameters, $\xi \sim \rho^\phi(\xi|u)$, with more samples in areas of the parameter space that are more likely to match the data-generating parameters ξ^*. We find that FUSE outperforms the other methods when sufficient input information is given. We observe that inVAErt experiences posterior collapse for PWP in this setting, which is represented in high CRPS (Figure A.9). Histograms reveal that FUSE captures the expected dependencies of the data on the parameters (Figures A.1, A.2), such as wider distributions for scarcer information in PWP. In the ACB case, sharp Dirac-like estimates are obtained

6

for most of the samples (Figure A.14). However, a good fit for the numerical model parameters seems to rely on an accurate estimate of the initial condition (Figure A.16).

Forward Problem The surrogate $\mathcal{G}^\theta(\xi)$ is tasked to assess the ability of FUSE to predict the continuous s from the data-generating parameters ξ^*. FUSE consistantly outperforms all other methods in both test cases. In particular, FUSE is able to reproduce very nonlinear dynamics and sharp velocity gradients in the ACB case (Figure A.15). Potential errors mostly lie in the bulk velocity of the anomaly, resulting in a stretching or squeezing of the time series at later times. As expected, predictions are slightly worse further away from the perturbation location, where the flow has become more turbulent, especially for the vertical velocity which exhibits strong oscillations (Figure A.21a). For the PWP, continuous predictions based on the true parameters have an L1 error of well below 0.3% at all locations (Figures A.6, A.7). Smaller arteries generally exhibit a slightly higher error variance, which fits the physical problem interpretation, as smaller diameters show more variations to pressure than larger system arteries.

Unified Uncertainty Propagation Given an input u, the concatenated evaluation of the inverse and forward parts of FUSE provides a distribution over the output functions s, by first sampling parameters ξ_i and then evaluating the forward emulator on this ensemble, yielding continuous samples s_i. This can be interpreted as the uncertainty introduced by simulating the data with the parameterized model FUSE was trained on: Uncertainty is introduced as the parameters are not fully identifiable in the inverse step, and is propagated and expanded onto the continuous output through the ensemble predictions on the parameter samples. Again, FUSE outperforms all other models significantly in all experiments in the L^1 and L^2 error over the mean predictions $\bar{s}$. Figures 2 and 3 show how FUSE captures both smooth and rough structures, even on the sample it performs worst on. It only exhibits slight difficulties in predicting a constant velocity close to zero for ACB. At each location of the ACB, the L^1 error obtained with the propagated ensemble mean $\bar{s}$ is comparable to the error by evaluating only the forward model, $s = \mathcal{G}^\theta(\xi^*)$ (Figure A.21b). An inspection of the predictions for all locations for the PWP case shows that the true time series lie within one standard deviation of the FUSE ensemble mean for the median sample, and within the min-max uncertainty bounds for the worst sample (Figures A.3, A.4). We observe that the accurate prediction of model parameters correlates to accurate predictions of pressures (Figure A.11).

| (a) FUSE | (b) UNet | (c) InVAErt | (d) cVANO | (e) GAROM* |

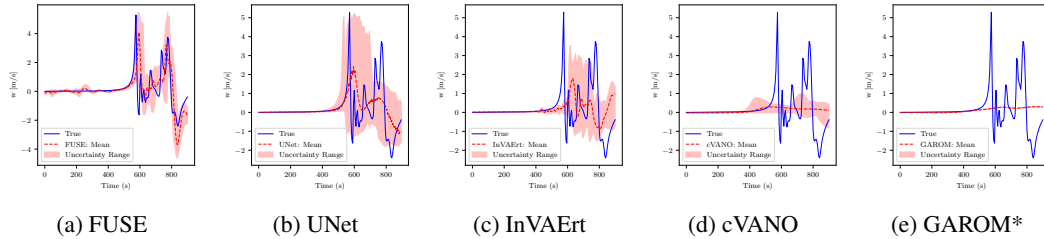



Figure 2: ACB, propagated uncertainty: Comparison to the benchmark models on the horizontal velocities at location 1, showing the unified ensemble predictions for the sample where FUSE has the largest L^1 error, i.e. a worst-case scenario. *GAROM predicts the velocity from the true parameters, while all other models predict pressure from the posterior distribution over parameters given continuous inputs.

| (a) FUSE | (b) UNet | (c) InVAErt | (d) cVANO | (e) GAROM* |

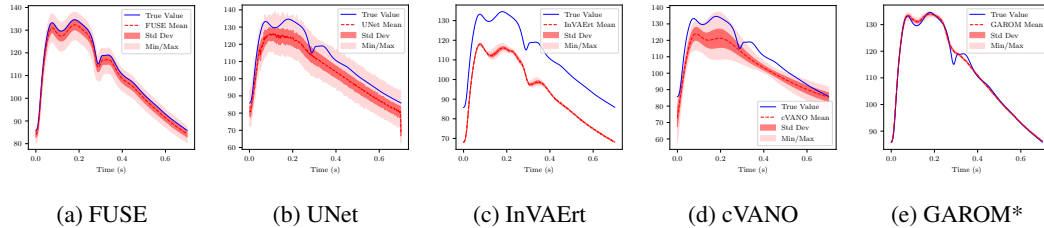



Figure 3: PWP, propagated uncertainty: Comparison to the benchmark models predicting the Sup. Middle Cerebral pressure and its uncertainty for the sample where FUSE has the largest L^1 error, i.e. a worst-case scenario. *see Figure 2.

7

Sensitivity Analysis Based on the good approximative properties showcased above, the forward operator $\mathcal{G}^\theta$ of FUSE can be used as a fast surrogate model to assess the sensitivity of the underlying PDE to varying the parameters ξ. Such analysis is usually constrained to a small sample size due to the large computational costs, and the dense sampling enabled by the surrogate helps to explore the parameter space for parameters that optimally fit patient data in PWP, or exhibit extreme winds in ACB. Varying one parameter at a time, while keeping all others fixed at their default value, exhibits the single effect on the data (*fingerprint*). For ACB, an increased amplitude of the cold anomaly mainly results in a speed-up and hence a squeezing of the time series, while other parameters have more nonlinear effects (Figure A.17). As the FUSE model provides no access to the full velocity fields, only numerical simulations reveal that these sharp sensitivities are mainly caused by certain features (eddies) of the flow reaching or missing a sensor depending on the parameter value. A validation of this sensitivity analysis for an estimate of the maximal horizontal velocity over the time series is provided in Figure 4 for a pairwise fingerprint with $10,000$ FUSE samples, with consistently low errors compared to 100 numerical samples. Only for one sample with a weak and small perturbation, the signal is not captured accurately. An additional evaluation of 60 samples at the margins of the parameter space shows that the peak velocities are consistently represented well (Figures A.20, A.19). For PWP, the sensitivity results (Figure A.10) are consistent with observations reported in the literature [7].

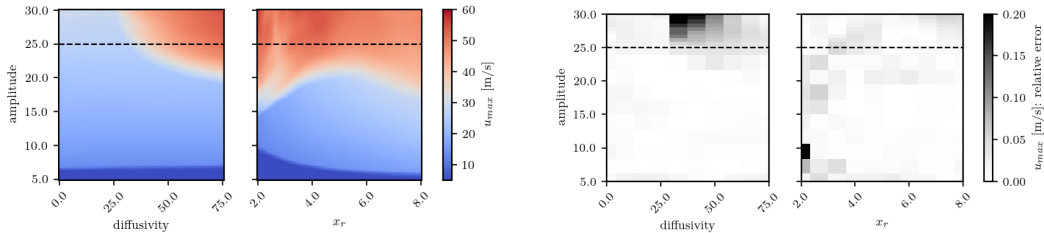



Figure 4: ACB, sensitivity analysis and generalization: Validation of the FUSE model against numeric simulations on peak horizontal velocities u at location 1 ($x = 15$ km, $z = 50$ m). Samples above the dashed line correspond to amplitudes larger than seen during training. The parameter resolution is 100×100 for FUSE, and $(10 + 10) \times 10$ for the numerical samples. Figure continued in the appendix, Figure A.18.

Out of Distribution Generalization for ACB We test the generalization capabilities of the FUSE forward model on test samples with amplitudes larger than seen in training in the pairwise sensitivity setup (Figure 4). The FUSE model is able to capture the shape of the pairwise parameter dependencies but struggles to locate the sharp dependence of the local peak velocity on the eddy diffusivity. Averaged errors in the out of distribution range are shown in Table 2, with FUSE performing clearly superior to all other models.

Levels of Available Information for PWP The case of missing input information is modeled by masking certain input components when evaluating the models. While FUSE shows regular performance comparable with the baselines on levels 2 and 3 of missing information, cVANO seems to be better suited for this setting (Table 1), probably due to the assumption that the latent dimensions are uncorrelated. As a sanity check, FUSE samples the prior distribution of the training data when evaluated without any conditional input information u (Figure A.8). When the available input measurement locations of the PWP experiment are known a priori, it is possible to get more accurate results when training FUSE without masking. For example, training to predict pressures and parameters from only the PPG data at the fingertip only, without masking, results in a CRPS of 4.28×10^{-2} and a relative L^1 error of 3.6%, roughly half the error of the masked model. If the measurement data is fixed, predictions without masking are much more suitable for analysis.

5 Discussion

Summary We propose FUSE, a framework unifying surrogate modeling and parameter identification for parametric PDEs by bridging operator learning with flow-matching posterior estimation. Both are represented in the FUSE objective through a deterministic forward loss and a probabilistic inverse loss, respectively. The joint architecture allows for inverse estimation of scalar parameters ξ given

8

9

Table 1: Performance of FUSE and the baseline models in estimating parameters ξ from continuous inputs u, quantified by CRPS, and predicting time series data s, quantified by a relative L_1 and L_2 error. Here, "True parameters" evaluates the forward model part only, and "Estimated parameters" and levels one to three evaluates the sample mean $\bar{s}$ predicted by the unified model.

	Model	Experimental Setup	CRPS$\times 10^2$	Rel. L_1 Error	Rel. L_2 Error
Pulse Wave Propagation	FUSE (**ours**)	True Parameters	-	**0.13 $\pm$ 0.05** %	**0.19 $\pm$ 0.08** %
		Level 1	**1.31 $\pm$ 0.69**	**0.86 $\pm$ 0.93** %	**0.90 $\pm$ 0.92** %
		Level 2	**3.18 $\pm$ 1.40**	**3.83 $\pm$ 2.66** %	**3.87 $\pm$ 2.64** %
		Level 3	8.48 $\pm$ 2.55	**6.56 $\pm$ 3.36** %	**7.05 $\pm$ 3.39** %
	UNet (**ablation**)	True Parameters	-	1.38 $\pm$ 0.34 %	2.00 $\pm$ 0.48 %
		Level 1	3.79 $\pm$ 0.97	3.81 $\pm$ 2.03%	4.25 $\pm$ 1.92 %
		Level 2	5.75 $\pm$ 1.32	6.79 $\pm$ 3.77 %	7.19 $\pm$ 3.68 %
		Level 3	12.72 $\pm$ 2.65	7.83 $\pm$ 2.64 %	8.78 $\pm$ 2.87 %
	InVAErt	True Parameters	-	3.44 $\pm$ 0.52 %	4.21 $\pm$ 0.60 %
		Level 1	11.62 $\pm$ 2.39	8.28 $\pm$ 4.78 %	8.94 $\pm$ 4.91 %
		Level 2	16.88 $\pm$ 3.67	7.85 $\pm$ 2.80 %	8.64 $\pm$ 2.99 %
		Level 3	21.22 $\pm$ 4.35	8.37 $\pm$ 2.73 %	9.48 $\pm$ 2.98 %
	cVANO	True Parameters	-	7.49 $\pm$ 2.60 %	8.51 $\pm$ 2.58 %
		Level 1	5.61 $\pm$ 0.24	7.54 $\pm$ 2.52 %	8.52 $\pm$ 2.53 %
		Level 2	4.00 $\pm$ 2.22	8.42 $\pm$ 3.11 %	9.72 $\pm$ 3.23 %
		Level 3	**3.66 $\pm$ 0.76**	10.79 $\pm$ 3.60 %	12.54 $\pm$ 4.05 %
	GAROM	True Parameters	-	0.70 $\pm$ 0.13 %	1.04 $\pm$ 0.22 %
		Levels 1-3	-	-	-
Atm. Cold Bubble	FUSE (**ours**)	True Parameters	-	**0.41 $\pm$ 0.45** %	**0.80 $\pm$ 0.89** %
		Estimated Parameters	**1.84 $\pm$ 1.63**	**0.58 $\pm$ 0.59** %	**1.13 $\pm$ 1.11** %
	UNet (**ablation**)	True Parameters	-	1.70 $\pm$ 0.90 %	2.86 $\pm$ 1.55 %
		Estimated Parameters	4.47 $\pm$ 2.67	2.13 $\pm$ 1.16 %	3.69 $\pm$ 1.94 %
	InVAErt	True Parameters	-	1.11 $\pm$ 0.85 %	1.94 $\pm$ 1.31 %
		Estimated Parameters	6.65 $\pm$ 9.05	1.17 $\pm$ 1.35 %	2.88 $\pm$ 2.04 %
	cVANO	True Parameters	-	5.32 $\pm$ 1.57 %	8.26 $\pm$ 2.44%
		Estimated Parameters	19.84 $\pm$ 4.73	8.77 $\pm$ 3.04 %	12.33 $\pm$ 4.00%
	GAROM	True Parameters	-	3.99 $\pm$ 1.67 %	6.35 $\pm$ 2.43 %
		Estimated Parameters	-	-	-

Table 2: Performance of FUSE on out-of-distribution samples for ACB, as described in Figure 4, following the format of Table 1.

	Model	Experimental Setup	CRPS$\times 10^2$	Rel. L_1 Error	Rel. L_2 Error
Out-of-distribution (ACB)	Fuse (**ours**)	True Parameters	-	**1.83 $\pm$ 1.10** %	**3.15 $\pm$ 1.80** %
		Estimated Parameters	**3.13 $\pm$ 1.13**	**1.63 $\pm$ 0.78** %	**2.77 $\pm$ 1.28** %
	UNet (**ablation**)	True Parameters	-	2.78 $\pm$ 0.32 %	4.37 $\pm$ 0.43 %
		Estimated Parameters	12.36 $\pm$ 5.47	3.43 $\pm$ 0.40 %	5.40 $\pm$ 0.58 %
	InVAErt	True Parameters	-	4.48 $\pm$ 2.65 %	6.49 $\pm$ 3.26 %
		Estimated Parameters	18.80 $\pm$ 19.09	4.31 $\pm$ 1.13 %	6.92 $\pm$ 1.83 %
	cVANO	True Parameters	-	9.31 $\pm$ 1.22 %	13.74 $\pm$ 1.54 %
		Estimated Parameters	24.76 $\pm$ 1.45	15.35 $\pm$ 1.28 %	21.20 $\pm$ 1.49%
	GAROM	True Parameters	-	7.22 $\pm$ 1.00 %	10.96 $\pm$ 1.22 %
		Estimated Parameters	-	-	-

continuous measurements u, as well as forward predictions of (other) continuous measurements s given ξ. Inheriting from the properties of its components, FUSE is discretization invariant with respect to the continuous inputs and outputs, and can represent arbitrary distributions in the fully interpretable parameter space. As this latent space is defined through a known parameterization of the underlying PDE, FUSE allows for accurate calibration and evaluation of a numerical solver, as well as performing downstream tasks such as parameter uncertainty quantification. On the parametric PDEs of pulse wave propagation (PWP) in the human cardiovascular system and an atmospheric cold bubble (ACB) with time series measurements, FUSE consistently outperforms four baseline methods designed to perform similar tasks. It also shows good out-of-distribution generalization and great flexibility when trained and evaluated on different levels of available input information.

Connection to Existing Methods The proposed methodology and the adapted implementation of the FNO and FMPE offer the following advantages over the existing methods for joint parameter estimation and forward emulation mentioned in the introduction. By incorporating Neural Operators in both the forward and inverse parts, we ensure discretization invariance throughout the model [3]. In contrast to Variational Autoencoders, we present a fully probabilistic formulation and require no restriction on the latent data distribution. This allows us to model arbitrary posterior distributions, whereas cVANO is restricted to Gaussian distributions. Likewise, FUSE can leverage powerful, existing neural operators which improves predictions on the output functions. The experiments have shown that FUSE excels over all baselines in both test cases. In particular, cVANO and GAROM fail to capture even rough characteristics of the flow in the forward emulation for the ACB case with nonlinear dynamics, and InVAErt shows a consistent bias towards the mean in the PWP task. Only in the case of missing information in the smoother PWP case, cVANO outperforms FUSE on the inverse problem. In terms of the choice of FNO in both the forward and inverse part of FUSE, the ablation with UNet layers exhibits larger uncertainties and worse performance of the predicted mean. An additional ablation, presented in the Appendix 5, shows that optimal transport probability paths perform better than the diffusion-based paths in a conditional DDPM model. Finally, Gaussian processes are often chosen for their ability to handle unstructured data. We incorporate this property by using the grid-independent FNO implementation by [34].

Applicability FUSE is explicitly formulated for finite-dimensional parameters and function measurements and outputs. This setting naturally arises when calibrating a numerical solver, with training data generated synthetically by the solver itself, and was demonstrated in two test cases. However, it is very common for real measured datasets, e.g. in bioengineering [22], and more specifically when involving PWP, to contain both time series data and vectors of parameters available for different patients. FUSE naturally extends to these datasets, where an underlying PDE is assumed but not formulated or simulated explicitly, including parameter inference using real data [59], precision medicine or solver calibration [46], and fingerprinting to discover parameter-disease correlations [55]. In the case that the parameters are non-physical, and hence not measurable, such as the numerical model parameters in the ACB case, this does of course not apply, and the main aim of solving the forward and inverse problem is to explore the solver itself. Moreover, FUSE may help to refine existing parameterizations either through disentanglement [18] or by identifying parameters with little influence on the simulated data.

Limitations and Future Work The FUSE framework aims to emulate a given parameterization of a PDE. If such a parameterization (or a corresponding dataset) is not available, VAE-based preprocessing [4] or manifold discovery [5] may be used. When applying FUSE to real measurements, the parameterization turns to be only an approximation of the dynamical system, whereas it is considered a perfect model in the experiments presented here. This structural model uncertainty has to be added to the interpretation of the unified uncertainties given by the FUSE model, as it has to be for any other inverse algorithm. In practice, measurements are always associated with a measurement error, which is considerable in particular for extreme measurements. The current implementation of FUSE does not allow for errors in the continuous inputs to be taken into account, and it is left for a future extension of the framework to include these into a fully Bayesian formulation of the inverse model. In spatially distributed applications, such as atmospheric modeling, the parameters of interest might be themselves space/time-dependent functions, such as surface properties or initial conditions [48]. Based on the scaling properties of the FMPE model for high-dimensional data such as images [35], FUSE is expected to scale well in size for high-dimensional parameter spaces, but is yet to be extended to a resolution-invariant function representation in latent space. These limitations will be addressed in future research.

10

Acknowledgments

G.K. would like to acknowledge the support from Asuera Stiftung via the ETH Zurich Foundation.

References

[1] J. Alastruey, K. H. Parker, S. J. Sherwin, et al. Arterial pulse wave haemodynamics. In *11th international conference on pressure surges*, volume 30, pages 401–443. Virtual PiE Led t/a BHR Group Lisbon, Portugal, 2012.

[2] A. Arzani, J.-X. Wang, M. S. Sacks, and S. C. Shadden. Machine learning for cardiovascular biomechanics modeling: challenges and beyond. *Annals of Biomedical Engineering*, 50(6): 615–627, 2022. doi:10.1007/s10439-022-02967-4.

[3] F. Bartolucci, E. de Bezenac, B. Raonic, R. Molinaro, S. Mishra, and R. Alaifari. Representation equivalent neural operators: a framework for alias-free operator learning. *Advances in Neural Information Processing Systems*, 36, 2024. URL https://papers.nips.cc/paper_files/paper/2023/hash/dc35c593e61f6df62db541b976d09dcf-Abstract-Conference.html.

[4] P. Bojanowski, A. Joulin, D. Lopez-Pas, and A. Szlam. Optimizing the latent space of generative networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 600–609. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/bojanowski18a.html.

[5] J. Brehmer and K. Cranmer. Flows for simultaneous manifold learning and density estimation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/051928341be67dcba03f0e04104d9047-Abstract.html.

[6] S. Čanić, J. Tambača, G. Guidoboni, A. Mikelić, C. J. Hartley, and D. Rosenstrauch. Modeling viscoelastic behavior of arterial walls and their interaction with pulsatile blood flow. *SIAM Journal on Applied Mathematics*, 67(1):164–193, 2006. doi:10.1137/060651562.

[7] P. H. Charlton, J. Mariscal Harana, S. Vennin, Y. Li, P. Chowienczyk, and J. Alastruey. Modeling arterial pulse waves in healthy aging: a database for in silico evaluation of hemodynamics and pulse wave indexes. *American Journal of Physiology-Heart and Circulatory Physiology*, 317 (5):H1062–H1085, 2019. doi:10.1152/ajpheart.00218.2019.

[8] P. Y. Chen, J. Xiang, D. H. Cho, Y. Chang, G. A. Pershing, H. T. Maia, M. M. Chiaramonte, K. T. Carlberg, and E. Grinspun. CROM: Continuous reduced-order modeling of PDEs using implicit neural representations. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/12094.

[9] E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *Journal of Computational Physics*, 424:109716, 2021. ISSN 0021-9991. doi:10.1016/j.jcp.2020.109716. URL https://www.sciencedirect.com/science/article/pii/S0021999120304903.

[10] D. Coscia, N. Demo, and G. Rozza. Generative adversarial reduced order modelling. *Sci Rep*, 14(3826), 2024. doi:10.1038/s41598-024-54067-z.

[11] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1410.8516.

[12] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real NVP. In *The Fifth International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

[13] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, volume 1. Springer Science & Business Media, 2010.

11

[14] M. Gloeckler, M. Deistler, C. D. Weilbach, F. Wood, and J. H. Macke. All-in-one simulation-based inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 15735–15766. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/gloeckler24a.html.

[15] M. Gulian, A. Frankel, and L. Swiler. Gaussian process regression constrained by boundary value problems. *Computer Methods in Applied Mechanics and Engineering*, 388:114117, 2022. doi:https://doi.org/10.1016/j.cma.2021.114117.

[16] R. Halder, M. Damodaran, and B. C. Khoo. Deep learning based reduced order model for airfoil-gust and aeroelastic interaction. *AIAA Journal*, 58(4):1595–1606, 8 2020. doi:10.2514/1.J059027.

[17] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song, and J. Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 12556–12569. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/hao23c.html.

[18] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *The Fifth International Conference on Learning Representations*, volume 3, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[19] S. Hijazi, M. Freitag, and N. Landwehr. POD-galerkin reduced order models and physics-informed neural networks for solving inverse problems for the navier–stokes equations. *Adv. Model. and Simul. in Eng. Sci.*, 10(5):5, 03 2023. doi:10.1186/s40323-023-00242-2.

[20] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://papers.nips.cc/paper_files/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[21] P. Jin, S. Meng, and L. Lu. MIONet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022. doi:10.1137/22M1477751.

[22] A. E. W. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016. doi:10.1038/sdata.2016.35.

[23] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. *Computer graphics forum*, 38(2):59–70, 2019. doi:10.1111/cgf.13619.

[24] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013. URL https://arxiv.org/abs/1312.6114.

[26] G. Kissas. *Towards Digital Twins for Cardiovascular Flows: A Hybrid Machine Learning and Computational Fluid Dynamics Approach*. PhD thesis, University of Pennsylvania, 2023.

[27] G. Kissas, J. H. Seidman, L. F. Guilhoto, V. M. Preciado, G. J. Pappas, and P. Perdikaris. Learning operators with coupled attention. *J. Mach. Learn. Res.*, 23(1), Jan. 2022. URL https://dl.acm.org/doi/10.5555/3586589.3586804.

[28] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL http://jmlr.org/papers/v24/21-1524.html.

12

[29] S. Lanthaler, S. Mishra, and G. E. Karniadakis. Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1): tnac001, 2022. doi:10.1093/imatrm/tnac001.

[30] M. Letafati, S. Ali, and M. Latva-aho. Conditional denoising diffusion probabilistic models for data reconstruction enhancement in wireless communications, 2024. URL https://arxiv.org/abs/2310.19460.

[31] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, volume 2010.08895, 2021. URL https://iclr.cc/virtual/2021/poster/3281.

[32] Z. Li, K. Meidani, and A. B. Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EPPqt3uERT.

[33] Z. Li, D. Shu, and A. Barati Farimani. Scalable transformer for pde surrogate modeling. In *Advances in Neural Information Processing Systems*, volume 36, pages 28010–28039. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/590daf74f99ee85df3d8c007df9c8187-Abstract-Conference.html.

[34] L. E. Lingsch, M. Y. Michelis, E. De Bezenac, S. M. Perera, R. K. Katzschmann, and S. Mishra. Beyond regular grids: Fourier-based neural operators on arbitrary domains. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 30610–30629. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lingsch24a.html.

[35] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/11309.

[36] S. Liu, C. Zeman, S. L. Sørland, and C. Schär. Systematic Calibration of a Convection-Resolving Model: Application Over Tropical Atlantic. *Journal of Geophysical Research: Atmospheres*, 127(23), 2022. doi:10.1029/2022JD037303.

[37] 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*, 3(3):218–229, Mar. 2021. doi:10.1038/s42256-021-00302-5.

[38] A. Moore. Investigating the Near-Surface Wind Fields of Downbursts using a Series of High-Resolution Idealized Simulations. *Weather and Forecasting*, 2024. doi:10.1175/WAF-D-23-0164.1.

[39] L. Orf, E. Kantor, and E. Savory. Simulation of a downburst-producing thunderstorm using a very high-resolution three-dimensional cloud model. *Journal of Wind Engineering and Industrial Aerodynamics*, 2012. doi:10.1016/j.jweia.2012.02.020.

[40] A. Parodi, M. Lagasio, M. Maugeri, B. Turato, and W. Gallus. Observational and Modelling Study of a Major Downburst Event in Liguria: The 14 October 2016 Case. *Atmosphere*, 10(12): 788, 2019. doi:10.3390/atmos10120788.

[41] O. Pauluis. Thermodynamic Consistency of the Anelastic Approximation for a Moist Atmosphere. *Journal of the Atmospheric Sciences*, 65(8), 2008. doi:10.1175/2007JAS2475.1.

[42] K. G. Pressel, C. M. Kaul, T. Schneider, Z. Tan, and S. Mishra. Large-eddy simulation in an anelastic framework with closed water and entropy balances. *Journal of Advances in Modeling Earth Systems*, 7(3):1425–1456, 2015. ISSN 1942-2466. doi:10.1002/2015MS000496. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/2015MS000496.

[43] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.

13

[44] B. Raonic, R. Molinaro, T. De Ryck, T. Rohner, F. Bartolucci, R. Alaifari, S. Mishra, and E. de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/f3c1951b34f7f55ffaecada7fde6bd5a-Abstract-Conference.html.

[45] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

[46] J. Richter, J. Nitzler, L. Pegolotti, K. Menon, J. Biehler, W. A. Wall, D. E. Schiavazzi, A. L. Marsden, and M. R. Pfaller. Bayesian windkessel calibration using optimized 0d surrogate models, 2024. URL https://arxiv.org/abs/2404.14187.

[47] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-24574-4_28.

[48] Y. Ruckstuhl and T. Janjić. Combined State-Parameter Estimation with the LETKF for Convective-Scale Weather Forecasting. *Monthly Weather Review*, 148(4):1607–1628, 2020. doi:10.1175/MWR-D-19-0233.1.

[49] P. Segers, E. Rietzschel, M. De Buyzere, N. Stergiopulos, N. Westerhof, L. Van Bortel, T. Gillebert, and P. Verdonck. Three-and four-element windkessel models: assessment of their fitting performance in a large cohort of healthy middle-aged individuals. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 222(4):417–428, 2008. doi:10.1243/09544119JEIM287.

[50] J. Seidman, G. Kissas, P. Perdikaris, and G. J. Pappas. NOMAD: Nonlinear manifold decoders for operator learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35. Curran Associates, Inc., 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/24f49b2ad9fbe65eefbfd99d6f6c3fd2-Abstract-Conference.html.

[51] J. H. Seidman, G. Kissas, G. J. Pappas, and P. Perdikaris. Variational autoencoding neural operators. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/seidman23a.html.

[52] S. Sherwin, L. Formaggia, J. Peiro, and V. Franke. Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system. *International Journal for Numerical Methods in Fluids*, 43(6-7):673–700, 2003. doi:10.1002/fld.543.

[53] Y. Shi, A. F. Gao, Z. E. Ross, and K. Azizzadenesheli. Universal functional regression with neural operator flows. In *Advances in Neural Information Processing Systems:Workshop on Bayesian Decision-making and Uncertainty*, 2024. URL https://openreview.net/forum?id=ZCtnWuaZiI.

[54] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[55] R. Sokolow, G. Kissas, C. Beeche, S. Swago, E. W. Thompson, M. Viswanadha, J. Chirinos, S. Damrauer, P. Perdakaris, D. J. Rader, and W. R. Witschey. An aortic hemodynamic fingerprint reduced order modeling analysis reveals traits associated with vascular disease in a medical biobank. *bioRxiv*, 2024. doi:10.1101/2024.04.19.590260.

14

[56] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier. Numerical solutions of a non-linear density current: A benchmark solution and comparisons. *International Journal for Numerical Methods in Fluids*, 17(1):1–22, 1993. doi:10.1002/fld.1650170103.

[57] G. G. Tong, C. A. Sing Long, and D. E. Schiavazzi. InVAErt networks: A data-driven framework for model synthesis and identifiability analysis. *Computer Methods in Applied Mechanics and Engineering*, 423:116846, 2024. doi:https://doi.org/10.1016/j.cma.2024.116846.

[58] R. Wang, Z. Chen, Q. Luo, and F. Wang. A conditional denoising diffusion probabilistic model for radio interferometric image reconstruction. In *Proceedings of the 11th International Conference on Artificial Intelligence and Applications (FAIA)*, 2023. doi:10.3233/FAIA230554.

[59] A. Wehenkel, J. Behrmann, A. C. Miller, G. Sapiro, O. Sener, M. C. Cameto, and J.-H. Jacobsen. Simulation-based inference for cardiovascular models. In *NeurIPS Workshop*, 2024. URL https://arxiv.org/abs/2307.13918.

[60] J. Wildberger, M. Dax, S. Buchholz, S. Green, J. H. Macke, and B. Schölkopf. Flow matching for scalable simulation-based inference. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc., 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/3663ae53ec078860bb0b9c6606e092a0-Abstract-Conference.html.

[61] J. Xiong, X. Cai, and J. Li. Clustered active-subspace based local gaussian process emulator for high-dimensional and complex computer models. *Journal of Computational Physics*, 450, 2022. doi:10.1016/j.jcp.2021.110840.

[62] J. Zhang, S. Zhang, and G. Lin. PAGP: A physics-assisted gaussian process framework with active learning for forward and inverse problems of partial differential equations, 2022. URL http://arxiv.org/abs/2204.02583.

15

A Supplementary Material

A.1 Training Details

All experiments were performed on an Nvidia GeForse RTX 3090 GPU with 24GB of memory. The number of training, validation, and test samples are provided in Table 3, along with the data dimensions. Further details regarding the experimental data are provided in Sections A.3 and A.4.

Training Strategy For each experiment, we performed a hyperparameter sweep when possible to select the learning rate, scheduler rate, weight decay, and network size. Due to the large number of hyperparameters, we selected 64 random combinations of hyperparameters and trained each model for 500 epochs. Following the sweep, we selected the model with the best performance and trained it for 2000 epochs, saving it at the epoch with the best performance on the validation set. Following this, we evaluated its performance on the test samples. A special exception must be made for the VAE-based models, InVAErt and cVANO. InVAErt was particularly susceptible to posterior collapse. Following the advice of the authors in Section 2.3.2 of [57], we hand-tuned the model to prevent over-parameterization and used early stopping. Similarly, cVANO was susceptible to training instability and exploding gradients, which also necessitated careful tuning and training in order to get the best results. The final size of each model, in number of network parameters, is provided in Table 4. The results from the model selection revealed all models are fairly consistent in size, typically all on the same order of magnitude.

We observed some sensitivity to the normalization scheme used on the data. In the PWP experiment, we normalize each input channel to a range between zero and one by dividing by the maximum value across all samples at that channel. In the ACB experiment, we normalize each channel by taking the minimum and maximum value across all samples in that channel and rescaling these to a range between zero and one.

The training time per epoch of all experiments is consistent, ranging from 1.5 to 2 seconds. During the hyperparameter sweep, each model is trained for 500 epochs, taking approximately 12.5 minutes. The best model from the sweep is trained for a total of 2000 epochs, taking approximately 1 hour. Throughout this entire project, we estimate the total number of GPU-hours for all model hyperparameter sweeps, hand-tuning, and training of the final model to be approximately 180 GPU-hours.

Additional Training Approaches Although the two objectives of FUSE are decoupled during training time, it is also possible to backpropagate losses on the outputs through both the forward and inverse model components. FMPE supports differentiable sampling, therefore the loss on the outputs s could be formulated as a function of continuous inputs u,

$$\mathcal{L}_3(\theta, \phi) = \mathcal{L}_1(\theta; \ \xi \sim \rho^\phi(.|u)) = \int_\mathcal{U} \left\| \mathcal{G}^\theta(\rho^\phi(\xi|u)) - \mathcal{G}(\rho(\xi^*|u)) \right\|_{L^1} \, \mathrm{d}u. \tag{9}$$

In the case where the inputs and outputs, respectively u and s, are identical, a *reconstruction* loss may be applied to further train the inverse model component. This entails predicting outputs s from the data-generating parameters ξ^* and predicting the inverse problem from the predictions s, as opposed to the original inputs u. This is formulated as

$$\mathcal{L}_r(\phi, \theta) = \mathcal{L}_2(\phi; \ \hat{u} = T \circ \mathcal{G}^\theta(\xi^*)) = \mathcal{L}^{FMPE}(\phi; \ \hat{u} = T \circ \mathcal{G}^\theta(\xi^*)). \tag{10}$$

Although each approach has the potential to aid training it is still required to train each objective separately. Likewise, we did not observe improvements in our experiments by implementing these additional losses. This further supports our claim in Equation 1, that model components trained by decoupled objectives for forward and inverse problems may be *unified* at evaluation to propagate uncertainties in the parameters to outputs and provide interpretable results for complex and dynamic systems of PDEs.

A.2 Evaluation Metrics

Continuous Ranked Probability Score We define the *Continuous Ranked Probability Score* (CRPS) and present how it compares a single ground truth value y to a Cumulative Distribution Function (CDF) F, quantifying how well a distribution of samples fits a single target. The CRPS is

16

Table 3: Data details. N_{train}, N_{val}, and N_{train} provide the number of training, validation, and test samples. The number of scalar parameters of the PDE system is given as m. "Channels" describe the number of continuous input components (locations/variables), and the number of points in the time discretization of the continuous inputs and outputs is given by "Sample Points". The "Batch Size" is the number of training samples per mini-batch during training.

Experiment	N_{train}	N_{val}	N_{test}	m	Channels	Sample Points	Batch Size
PWP	4000	128	128	32	52	487	64
ACB	8000	1000	1000	6	20	181	64

Table 4: Number of network parameters (millions) per model for the experiments.

Model	PWP	ACB
FUSE (**ours**)	7.2	5.8
UNet	14.1	9.8
cVANO	5.7	1.8
GAROM	11.1	2.7
InVAErt	13.6	2.9

Table 5: Inference times and DDPM ablation. Inference time, CRPS, and relative L1 errors over continuous outputs for a trained model are evaluated as the number of FMPE and Conditional DDPM samples increase. Conditional DDPM is implemented as described [30] for a 1D problem using 32 diffusion steps. We observe that the FMPE method results to higher accuracy in both metrics and slight increases in accuracy up until 512 samples, while DDPM is fairly consistent. Sampling times are approximately equivalent between the two models.

	Posterior Samples	Forw. Pass (s)	Inverse Pass (s)		CRPS$\times 10^2$		L_1 Error	
		FNO	FMPE	DDPM	FMPE	DDPM	FMPE	DDPM
case: ACB	128	0.0018	0.179	0.108	1.84 ± 1.63	4.96 ± 2.23	0.58 ± 0.58	1.61 ± 0.91
	256	0.0030	0.194	0.114	1.64 ± 1.63	4.75 ± 2.25	0.58 ± 0.58	1.60 ± 0.90
	512	0.0059	0.211	0.114	1.54 ± 1.64	4.61 ± 2.16	0.58 ± 0.58	1.60 ± 0.90
	1024	0.0127	0.240	0.113	1.49 ± 1.64	4.57 ± 2.20	0.58 ± 0.58	1.59 ± 0.89
	2048	0.0239	0.495	0.152	1.46 ± 1.64	4.54 ± 2.21	0.58 ± 0.58	1.59 ± 0.89
	4096	0.0608	0.973	0.285	1.45 ± 1.64	4.53 ± 2.21	0.58 ± 0.58	1.60 ± 0.90
	8192	0.1205	1.820	0.550	1.44 ± 1.64	4.52 ± 2.20	0.58 ± 0.58	1.59 ± 0.89

Table 6: Evaluating training time, CRPS, and relative L1 errors for models as the number of training data increases. As more training samples are used, CRPS and relative L1 error improve; however, the model converges after several thousand training samples resulting in smaller performance gains per new training sample.

	Training Data	Training Time Per Epoch (s)	CRPS$\times 10^2$	L_1 Error
case: ACB	128	0.027	16.10 ± 4.53	5.70 ± 2.34
	256	0.059	12.61 ± 4.85	4.22 ± 2.29
	512	0.118	10.23 ± 4.31	2.51 ± 1.60
	1024	0.213	8.48 ± 4.20	1.71 ± 1.30
	2048	0.450	5.88 ± 3.47	1.13 ± 0.95
	4096	0.973	3.90 ± 2.52	0.84 ± 0.71
	8192	1.698	1.84 ± 1.63	0.58 ± 0.60

defined as

$$CRPS(F, y) = \mathbb{E}\left[|X - y|\right] - \frac{1}{2} \cdot \mathbb{E}\left[X - X'\right], \tag{11}$$

17

with independent and identically distributed random variables X and X' following the distribution dictated by F. The first expectation computes the mean absolute error of the samples compared to the target value, and the second expectation penalizes tight distributions which are far from the true value. A CRPS of zero indicates that the distribution perfectly fits the target sample by returning a Dirac measure.

L^p **Error** We employ the relative L^1 and L^2 error for evaluating predictions of time series data. This metric is calculated by

$$\frac{||\mathcal{G}(\xi) - \mathcal{G}^\theta(\xi)||_p}{||\mathcal{G}(\xi)||_p}, \tag{12}$$

where $\mathcal{G}(\xi)$ is the true function described by the system of PDEs, and $\mathcal{G}^\theta(\xi)$ is a prediction produced by the network.

18

A.3 Pulse Wave Propagation: Reduced Order Model for Blood Flow

The reduced order Navier-Stokes model considers a system of discrete compliant tubes and arterial segments, connected at points [1]. The length of the vessels is considered much larger than the local curvature and the wave propagation happens on the axial direction. Therefore, their properties can be described using a Cartesian coordinate x [52]. The luminal cross-sectional area is defined as $A(x,t) = \int_A d\sigma$, the average velocity is defined as $U(x,t) = \frac{1}{A}\int_S \mathbf{u}(\mathbf{x},t) \cdot \mathbf{n}d\sigma$ and the volume flux at a given cross section as $Q(x,t) = UA$ [1]. The arteries are allowed to deform in the radial direction due to the internal pressure $P(x,t)$ which is considered constant over a cross section and the wall is considered impermeable. The blood is considered as an incompressible Newtonian fluid, with viscosity $\mu = 2.5mPa\ s$ and density $\rho = 1060kg\ m^{-3}$ [13]. All the gravitational effects are ignored assuming a patient in a supine position [1]. Using the above assumptions, the pulse wave propagation is formulated as the following system of hyperbolic constitutive laws [1, 13]:

$$\frac{\partial A}{\partial t} + \frac{\partial AU}{\partial x} = 0,$$
$$\frac{\partial U}{\partial t} + (2\alpha - 1)U\frac{\partial U}{\partial x} + (\alpha - 1)\frac{U^2}{A}\frac{\partial A}{\partial x} + \frac{1}{\rho}\frac{\partial P}{\partial x} = \frac{f}{\rho A},$$

(13)

where $a(x,t) = \frac{1}{AU^2}\int_A \mathbf{u}^2 d\sigma$ the Coriolis coefficient that accounts for the non-linearity of the integration over the cross section and describes the shape of the velocity profile. The velocity profile is commonly considered as axisymmetric, constant, and satisfying the non-slip boundary conditions on the arterial walls. An example of such profile is [1]:

$$u(x,r,t) = U\frac{\zeta + 2}{\zeta}\left[1 - (\frac{r}{R})^\zeta\right]$$

(14)

where r is the radial coordinate, $R(x,t)$ the lumen radius and $\zeta = \frac{2-\alpha}{\alpha - 1}$ a constant. A flat profile is defined by $\alpha = 1$ [7]. Integrating the three-dimensional Navier–Stokes equations for the axisymmetric flow provides the friction force per unit length $f(x,t) = -2(\zeta + 2)\mu\pi U$. To account for the fluid-structure interaction of the problem, a pressure area relation is derived by considering the tube law. For a thin isotropic homogenous and incompressible vessel that deforms axisymmetrically at each circular cross section the following Voigt-type visco-elastic law [6] is typically used:

$$P = P_e(A; x) + \frac{\Gamma(x)}{A_0(x)\sqrt{A}}\frac{\partial A}{\partial t},$$
$$P_e(A, x) = P_{\text{ext}} + \frac{\beta(x)}{A_0(x)}\left(\sqrt{A} - \sqrt{A_0(x)}\right),$$
$$\beta(x) = \frac{4}{3}\sqrt{\pi}E(x)h(x), \quad \Gamma(x) = \frac{2}{3}\sqrt{\pi}\phi(x)h(x),$$

(15)

where P_e is the elastic pressure component, $h(x)$ the wall thickness, $E(x)$ the Young's modulus, $\phi(x)$ the wall viscosity, $\beta(x)$ a wall stiffness coefficient, and $\Gamma(x)$ the wall viscosity. The scalars P_{ext} and A_0 correspond to the external pressure and equilibrium cross-sectional area, respectively. The parameters $\beta(x)$ and $\Gamma(x)$ are computed using empirical relations found in the literature [7]. For this problem, a pulse wave from the heart is considered as a boundary condition for the inlet of the artery. For merging and splitting of arteries, the conservation of mass and the continuity of dynamic and kinematic pressure are considered, as well as no energy losses. For the outlet of the arterial system, boundary conditions that model the downstream circulation are considered. In [7], the three-element Windkessel model [49] is used:

$$Q(1 + \frac{R_1}{R_2}) + CR_1\frac{\partial Q}{\partial t} = \frac{P_e - P_{\text{out}}}{R_2} + C\frac{P_e}{t}$$

(16)

In [7], the authors consider algebraic relations from the literature that allow them to include different parameters, e.g. the volume flux, in the pulse wave propagation problem. For example, the digital PPG measurements can be related to the volume flux at the extremities as:

$$\text{PPG}(t) = \int_0^T Q_W(t) - Q_{\text{out}}(t)\mathrm{d}t,$$

(17)

where T is the duration of a cardiac cycle, Q_W the flow entering the Windkessel model, and Q_{out} the flow in the outlet.

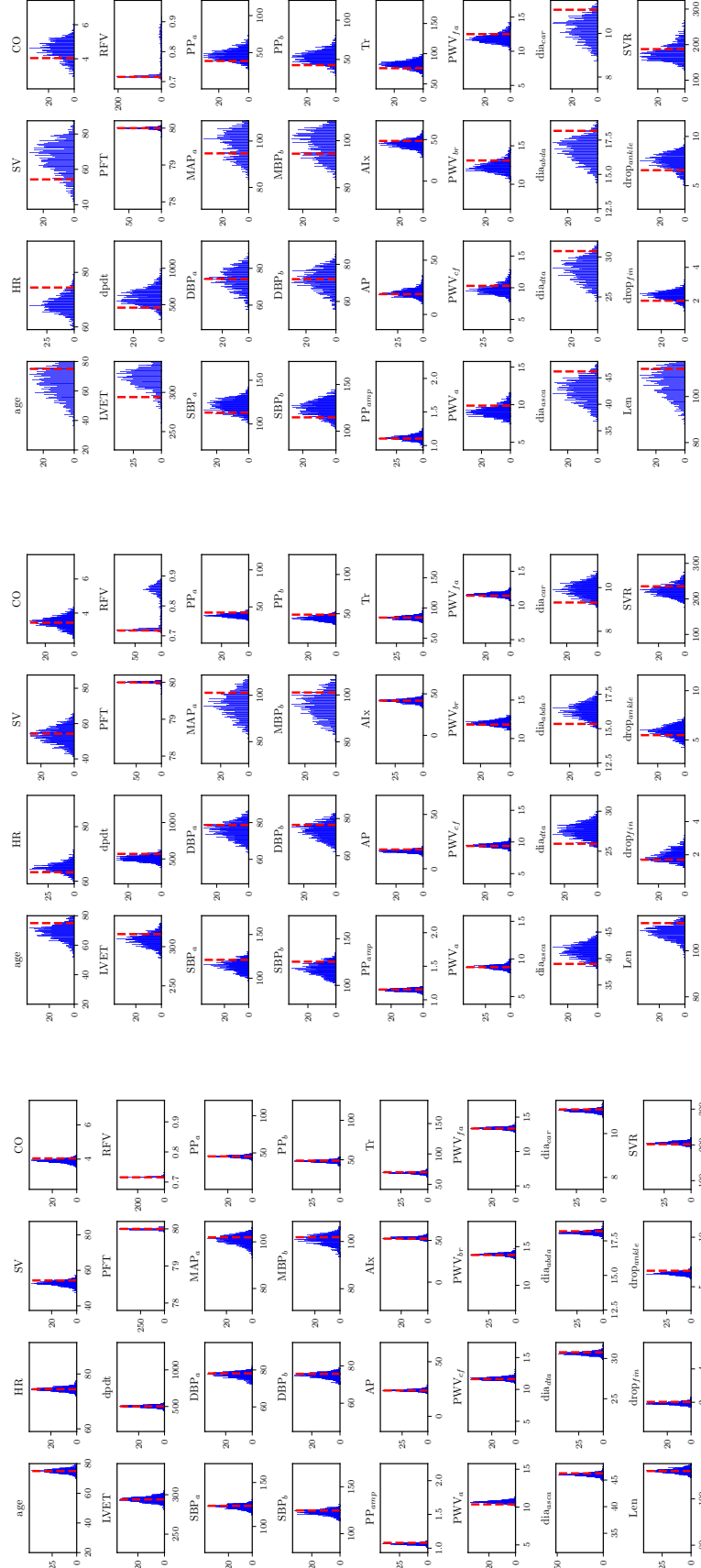
19

20

Table 7: Cardiovascular parameters used in the reduced order model for the blood flow pulse wave propagation.

Parameter	Units	Description
Age	years	Age of the individual
HR	bpm	Heart Rate
SV	ml	Stroke Volume
CO	l/min	Cardiac Output
LVET	ms	Left Ventricular Ejection Time
dp/dt	mmHg/s	Rate of pressure change in the heart
PFT	ms	Pulse Transit Time
RFV	ml	Residual Filling Volume
SBP_a	mmHg	Systolic Blood Pressure (arterial)
DBP_a	mmHg	Diastolic Blood Pressure (arterial)
MAP_a	mmHg	Mean Arterial Pressure
PP_a	mmHg	Pulse Pressure (arterial)
SBP_b	mmHg	Systolic Blood Pressure (brachial)
DBP_b	mmHg	Diastolic Blood Pressure (brachial)
MBP_b	mmHg	Mean Blood Pressure (brachial)
PP_b	mmHg	Pulse Pressure (brachial)
PP_{amp}	ratio	Pulse Pressure Amplification
AP	mmHg	Augmentation Pressure
AIx	%	Augmentation Index
Tr	ms	Reflection Time
PWV_a	m/s	Pulse Wave Velocity (arterial)
PWV_{cf}	m/s	Pulse Wave Velocity (carotid-femoral)
PWV_{br}	m/s	Pulse Wave Velocity (brachial-radial)
PWV_{fa}	m/s	Pulse Wave Velocity (femoral-ankle)
dia_{asca}	mm	Diameter of Ascending Aorta
dia_{dta}	mm	Diameter of Descending Thoracic Aorta
dia_{abda}	mm	Diameter of Abdominal Aorta
dia_{car}	mm	Diameter of Carotid Artery
Len	mm	Length
$drop_{fin}$	mmHg	Pressure Drop at Finger
$drop_{ankle}$	mmHg	Pressure Drop at Ankle
SVR	10^6 Pa s / m^3	Systemic Vascular Resistance

Table 8: Artery locations used in the reduced order model for the blood flow pulse wave propagation.

Artery Location	Description
Aortic Root	The portion of the aorta that is attached to the heart
Carotid	Major arteries in the neck leading to the brain, neck, and face
Thoracic Aorta	The part of the aorta that runs through the chest
Brachial	The major artery of the upper arm
Radial	Artery located in the forearm, commonly used to measure the pulse
Abdominal Aorta	The part of the aorta that runs through the abdomen
Iliac Bifurcation	The point where the aorta splits into the common iliac arteries
Common Iliac	Arteries that supply the pelvic organs and lower limbs
Femoral	The major artery supplying blood to the thigh and lower limbs
Anterior Tibial	Artery located in the lower leg
Sup. Middle Cerebral	Artery that supplies blood to the cerebral hemispheres
Sup. Temporal	Artery that supplies blood to parts of the face and scalp
Digital	Arteries that supply blood to the fingers

(a) Information level 1

(b) Information level 2

(c) Information level 3

Figure A.1: PWP, inverse problem: Histograms for the sample with the median CRPS for each level of available input information.

21

(a) Information level 1

(b) Information level 2

(c) Information level 3

Figure A.2: PWP, inverse problem: Histograms for the sample with the greatest CRPS (worst case) for each level of available input information.

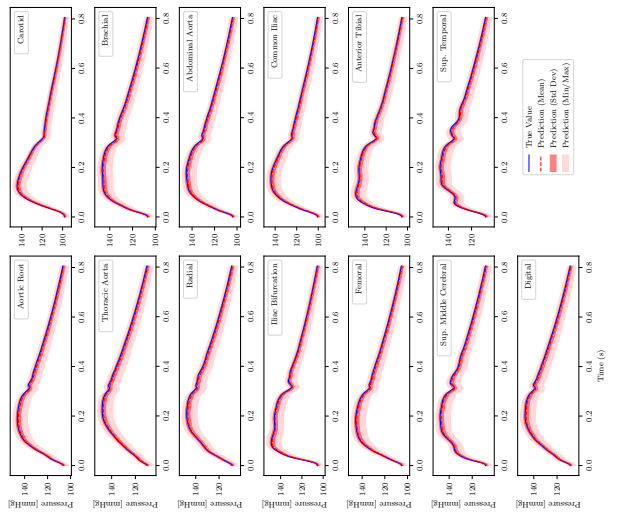
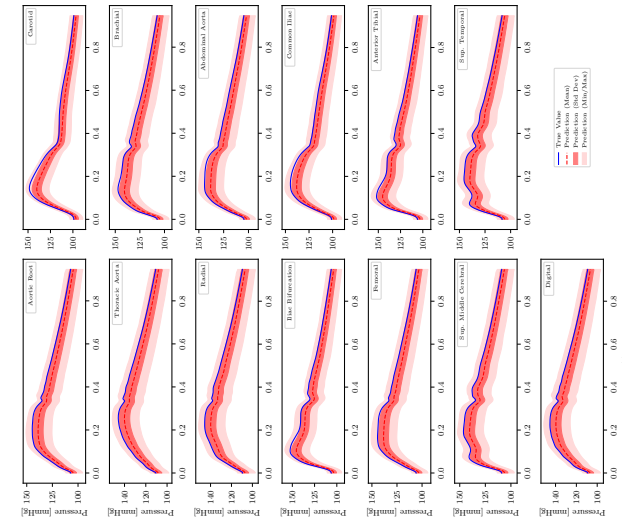
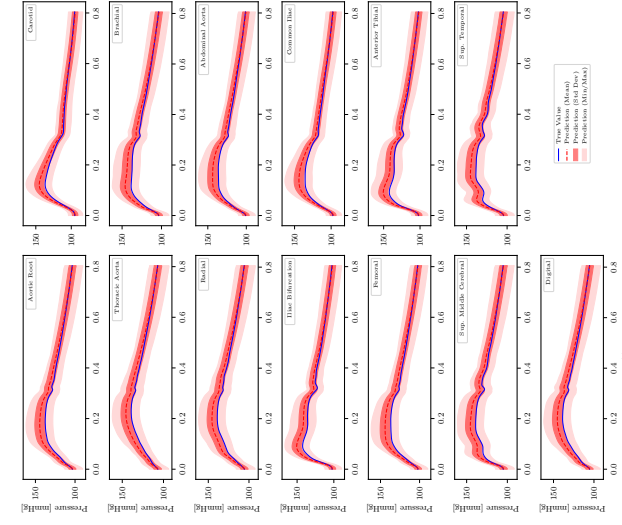
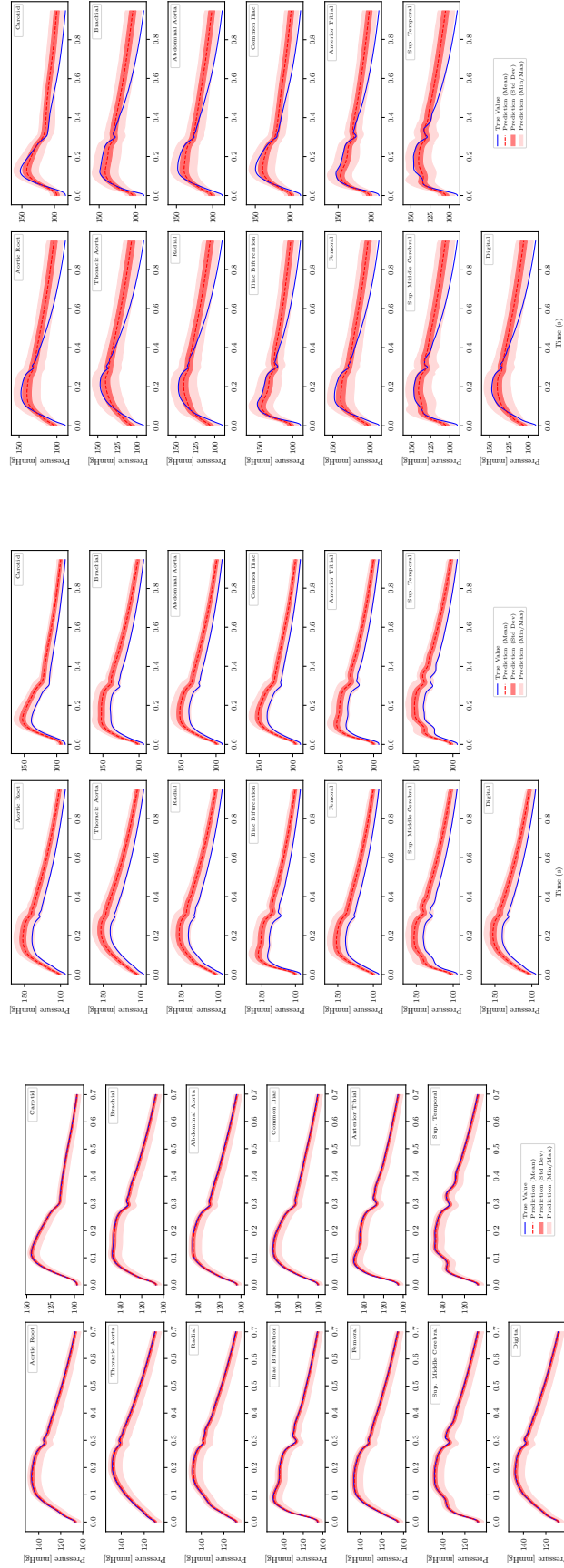
22

(a) Information level 1

(b) Information level 2

(c) Information level 3

Figure A.3: PWP, propagated uncertainty: Pressure time series for the sample with the median rel. $\mathcal{L}_1$ Error for each level of available input information.

23

(a) Information level 1

(b) Information level 2

(c) Information level 3

Figure A.4: PWP, propagated uncertainty: Pressure time series for the sample with the greatest rel. $\mathcal{L}_1$ Error (worst case) each level of available input information.

24

25

(a) CRPS on the inverse problem. (b) L^1 error on the forward and unified problem.

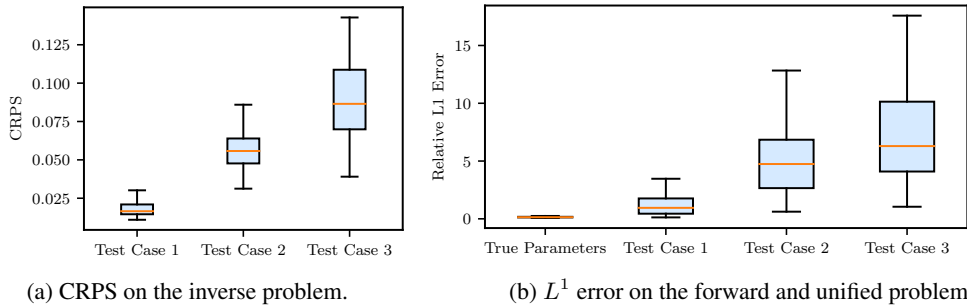



Figure A.5: PWP, performance summary: Box plots of the errors of the forward and inverse components of FUSE, as reported in Table 1, for different levels of available input information ("test cases"). For the inverse problem, parameter samples ξ_i are sampled from $\rho^\phi(\xi^i|u)$. For the forward problem, the output function s is predicted based on the true parameter values $\xi^* \sim \rho(\xi|u)$. For the unified problem evaluating both the inverse and forward model parts, the mean $\bar{s}$ of the ensemble prediction s_i from inferred parameters $\xi_i \sim \rho^\phi(\xi^i|u)$ is compared to the true output time series s. As information is removed from the input in the different cases, it becomes more difficult to estimate s.

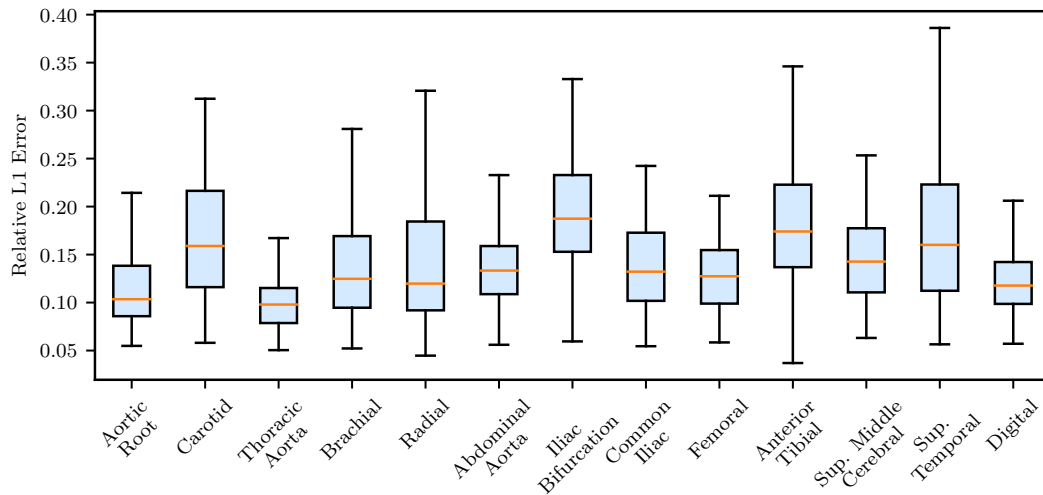



Figure A.6: PWP, propagated uncertainty: Relative L^1 Error per vessel predicting pressure time series from the true parameters.

26

Figure A.7: PWP, forward problem: Predictions with the largest L^1 error using the true parameters of the cardiovascular model.

Figure A.8: PWP, inverse problem: In the case of a fully masked input (all waveforms are zero), FUSE (blue) returns distributions which closely match the training (prior) distributions (orange).

27

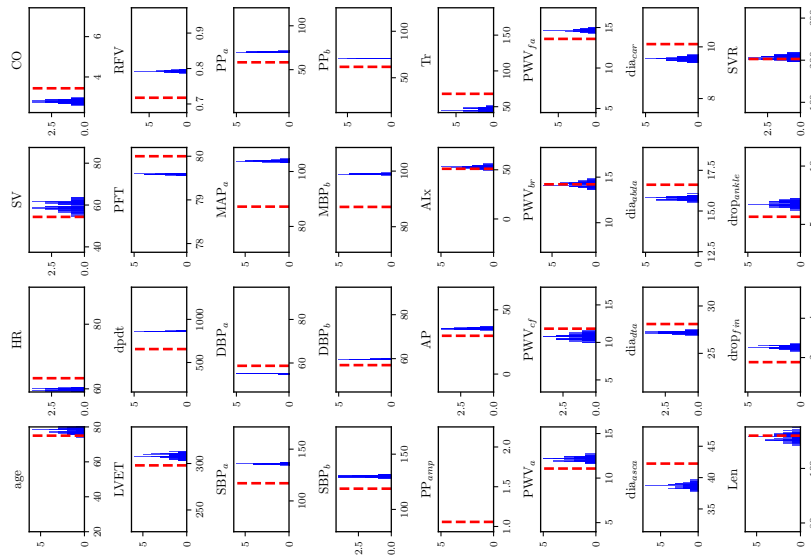



Figure A.9: PWP, inverse problem: InVAErt predictions over parameters, the median sample with full input information (level 1). InVAErt is particularly susceptible to posterior collapse, i.e., the predictions over the parameters become overconfident and the uncertainty bounds fail to capture the true parameter.

(a) Age: 25 (blue) to 65 (red)

(b) DBP: 55 (blue) to 90 (red)

(c) HR: 60 (blue) to 90 (red)

(d) SBP: 80 (blue) to 165 (red)

(e) LVET: 230 (blue) to 360 (red)

(f) MAP: 75 (blue) to 125 (red)

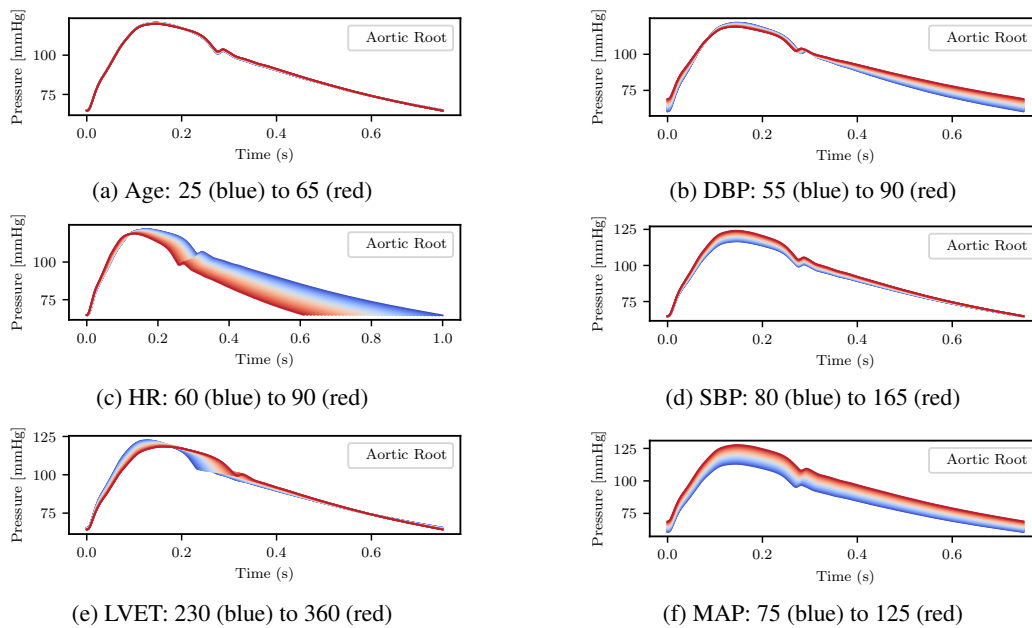



Figure A.10: PWP, sensitivity analysis (fingerprint): Pressure in the Aortic Root. We study how the pressure changes due to parameters of the system. Blue represents the lower bound of possible values, and red represents the upper bound, with colors of varying hue at intermediate values.

28

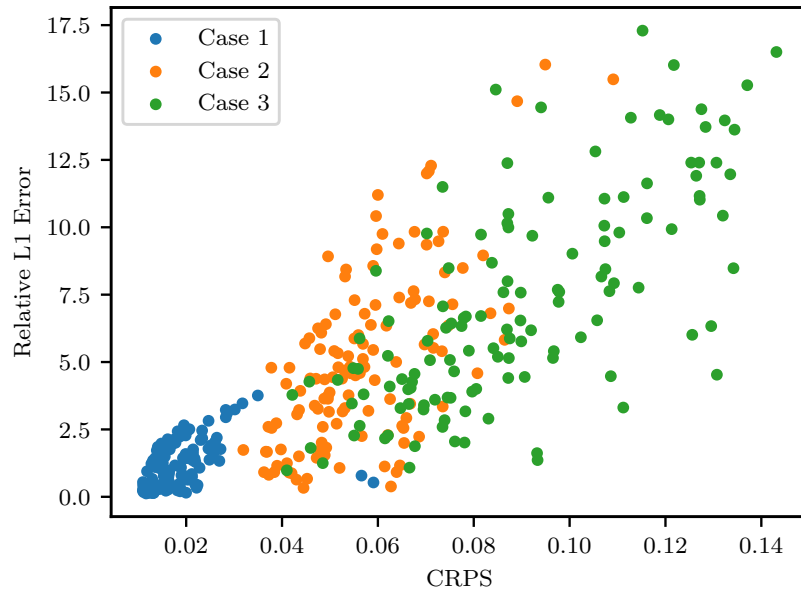



Figure A.11: PWP: Correlation of the errors between function outputs (L^1) and finite-dimensionalparameters (CRPS)for different levels of available input information ("levels", or "cases").

A.4 Atmospheric Cold Bubble

In atmospheric modelling, large-eddy simulations (LES) help to understand the turbulent dynamics within convective storms [39] that are usually observed only by scalar local measurements of wind (anemometers), other scalar measurements obtained at surface-level weather stations (e.g., pressure, humidity), and, if available, three-dimensional wind lidar records. A particularly destructive phenomenon associated with storms are *downbursts*, sudden local downdrafts that cause extreme winds when hitting the surface. Manual tuning of a numerical model is usually required to fit a simulation to match an observed downburst (e.g., [40]), due to the model complexity involving several parameterizations such as cloud microphysics. The model can then be configured to explore the sensitivity of the extreme winds both to the numerical model parameters and to the physical environmental conditions [38]. In addition to improving the scientific understanding and forecasts for more precise warnings, simulations of downbursts help to design robust infrastructure, such as wind turbines. However, the amount of simulations performed for model tuning (inverse problem) and sensitivity analysis (forward problem) is limited by the high computational cost to typically tens to hundreds of simulations. The affordable number of numerical simulations is either used for strategic expert tuning or to train an emulator for the parameter-to-data map, which is cheaper to evaluate and can then be used for in-depth sensitivity analysis (e.g., Gaussian Processes [9]; quadratic regression [36]). We would like to highlight that the presented test case is conceptually similar to downbursts, but far from simulating real-world scenarios due to its idealized nature and reduced complexity. However, we believe that test cases like this one will help to pave the way for handling realistic cases with methodologies such as FUSE.

Model Description The numerical data for the cold bubble simulation is obtained with the PyCLES model [42]. It solves the anelastic Navier–Stokes equations as formulated by [41], evolving around a hydrostatic reference state,

$$\alpha_0 \frac{\partial p_0}{\partial x_3} = -g, \tag{18}$$

with reference profiles $\alpha_0(z)$ for specific volume and $p_0(z)$ for pressure, and $(x_1, x_2, x_3) = (x, y, z)$ and $(u_1, u_2, u_3) = (u, v, w)$ for simplicity. The anelastic equations of motion are given by

$$\begin{cases} \dfrac{\partial u_i}{\partial t} + \dfrac{1}{\rho_0} \dfrac{\partial(\rho_0 u_i u_j)}{\partial x_i} = -\dfrac{\partial \alpha_0 p'}{\partial x_i} + b\delta_{13} - \dfrac{1}{\rho_0} \dfrac{\partial(\rho_0 \tau_{ij})}{\partial x_j} + \Sigma_i, & \text{(19a)} \\[3mm] \dfrac{\partial s}{\partial t} + \dfrac{1}{\rho_0} \dfrac{\partial(\rho_0 u_i s)}{\partial x_i} = \dfrac{Q}{T} - \dfrac{1}{\rho_0} \dfrac{\partial(\rho_0 \gamma_{s,i})}{\partial x_i} + \dot{S}, & \text{(19b)} \\[3mm] \dfrac{\partial \rho_0 u_i}{\partial x_i} = 0, & \text{(19c)} \end{cases}$$

encompassing the momentum equations (a), the entropy equation (b), and the continuity equation (c). We use standard conventions for summing and the Kronecker delta δ_{ij}, and left out the terms and further equations referring to the Coriolis force and moisture, as they do not apply to the small-scale and dry cold bubble simulation. The buoyancy term b depends on the specific volume, moisture, gravity, and the reference pressure profile p_0, p' is the dynamic pressure perturbation over the hydrostatic reference state, and Σ_i is an additional momentum source term, which is expected to be small for the cold bubble experiment. In the entropy equation, Q is the diabatic heating rate, T is the temperature, and $\dot{S}$ is a source term of irreversible entropy sources. Finally, τ_{ij} and $\gamma_{s,i}$ are the sub-grid scale (SGS) stresses acting on velocity and entropy, which model the effect of turbulence smaller than the grid scale in a diffusive way as

$$\tau_{ij} = -2\nu_t S_{ij}, \tag{20}$$

relating the sub-grid stress to the strain rate $S_{ij} = 1/2(\partial_i u_j + \partial_j u_i)$ of the resolved flow. The eddy viscosity ν_t may be related to the diffusivity D_t for heat and other scalars ϕ, and ultimately to the corresponding SGS stresses $\gamma_{\phi,i}$ by the turbulent Prandtl number Pr_t by

$$\begin{aligned} D_t &= \nu_t/\mathrm{Pr}_t, \\ \gamma_{\phi,i} &= -D_t \frac{\partial \phi}{\partial x_i}. \end{aligned} \tag{21}$$

For three-dimensional simulations, one may choose a sub-grid scale parameterization for ν_t such as a Smagorinsky-type first-order closure or a higher-order closure based on a prognostic equation for

29

Table 9: ACB: Discrete parameters with their ranges used for uniform sampling of the training data, as well as default values used by [56]. The parameters either encode the initial condition (IC) or are part of the sub-grid scale (SGS) model (M). The horizontal location x_c is kept fixed to ensure the horizontal symmetry of the domain.

Parameter	Name	Unit	Type	min	default	max
x_c	horizontal location	km	IC	-	**26.2**	-
x_r	horizontal radius	km	IC	2	**4**	8
z_c	vertical location	km	IC	2.5	**3**	3.5
z_r	vertical radius	km	IC	1	**2**	2.5
a	amplitude	K	IC	5	**15**	25
ν_t	eddy viscosity	m^2/s	M	0	**75**	75
D_t	eddy diffusivity	m^2/s	M	0	**75**	75

turbulent kinetic energy. For the two-dimensional cold bubble case, however, we stick to uniform values for both ν_t and D_t, independent of the turbulent Prandtl number.

The domain of size $(L_x, \; L_z) = (51.6 \text{ km}, \; 6.4 \text{ km})$ exhibits a neutrally stratified background profile ($\partial_z \theta = 0$, where θ denotes potential temperature), and no background winds. An elliptic cold air anomaly is prescribed by

$$
\begin{aligned}
\Delta T &= -a \left(\cos(\pi L) + 1 \right) / 2, \\
L &= \left\| \left((x - x_c) x_r^{-1}, \; (z - z_c) z_r^{-1} \right) \right\|_2,
\end{aligned}
\tag{22}
$$

where default values and ranges for all parameters can be found in Table 9, with the resulting flow fields shown in Figure A.12.

Numerical Setup The PyCLES model by [42] relies on a WENO finite volumes scheme that exhibits exceptional stability, implemented in Cython. It is used here in the setup described by the authors, at resolution $\Delta x = \Delta z = 50$ m and an adaptive time stepping bound to $\Delta t \leq 5$ s in order to record time series measurements at a temporal resolution of five seconds.

The data is recorded at the eight locations marked in Figure A.12, at horizontal locations $x = 15, \; 20$ km (about 10 and 5 km from the center of the anomaly), and vertical heights of $z = 50, \; 100, \; 250, \; 500$, and 2000 m, each being part of the numerical grid.

In terms of computational complexity, a PyCLES simulation in the given setup takes about half an hour on eight cores.

30

31

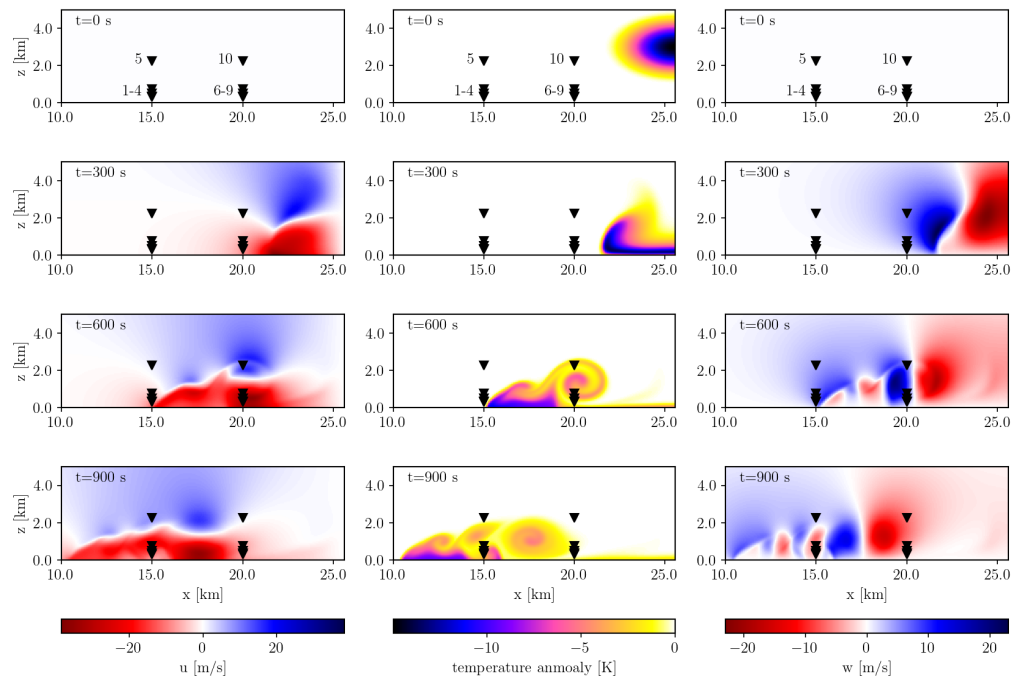



Figure A.12: ACB: Time evolution of horizontal velocity (left), temperature anomaly (middle), and vertical velocity (right), for the default values of the parameters given in Table 9, showing three main vortices that develop until $T = 900$ s. Measurement locations for the time series data are marked with triangles, with their location indices indicated in the first row.

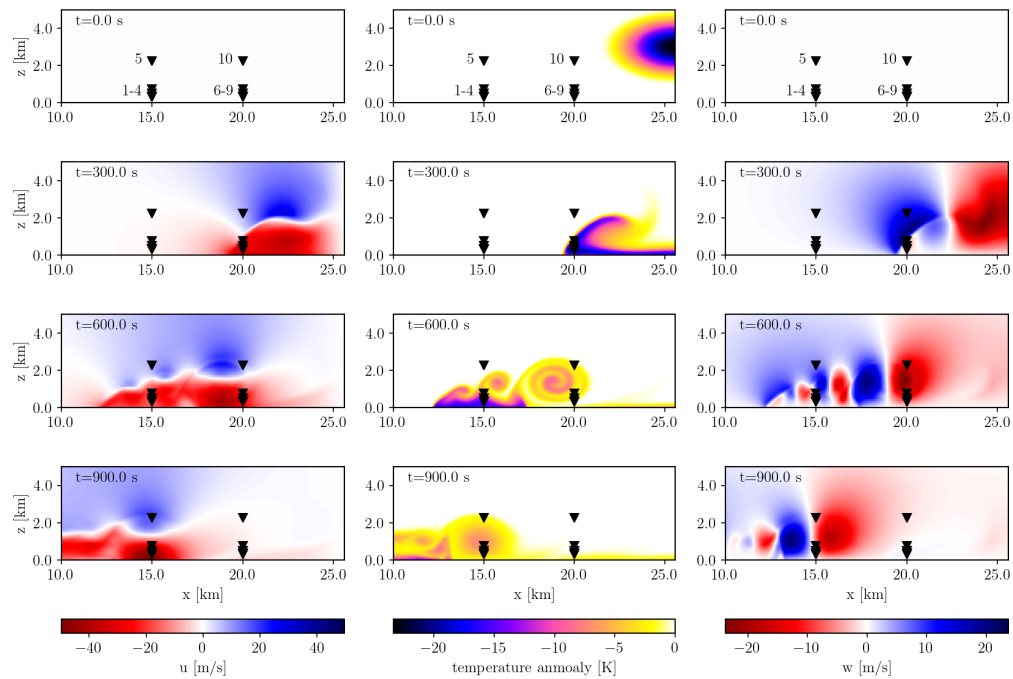
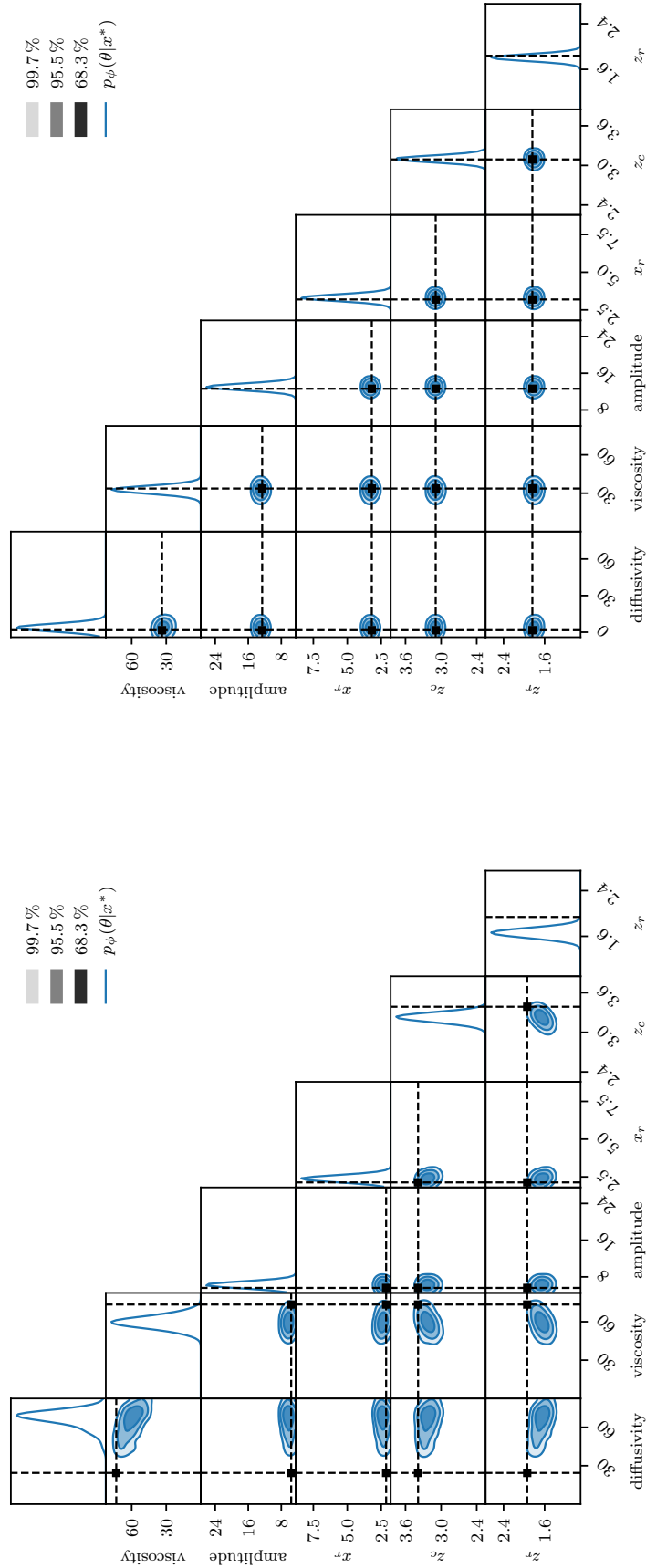
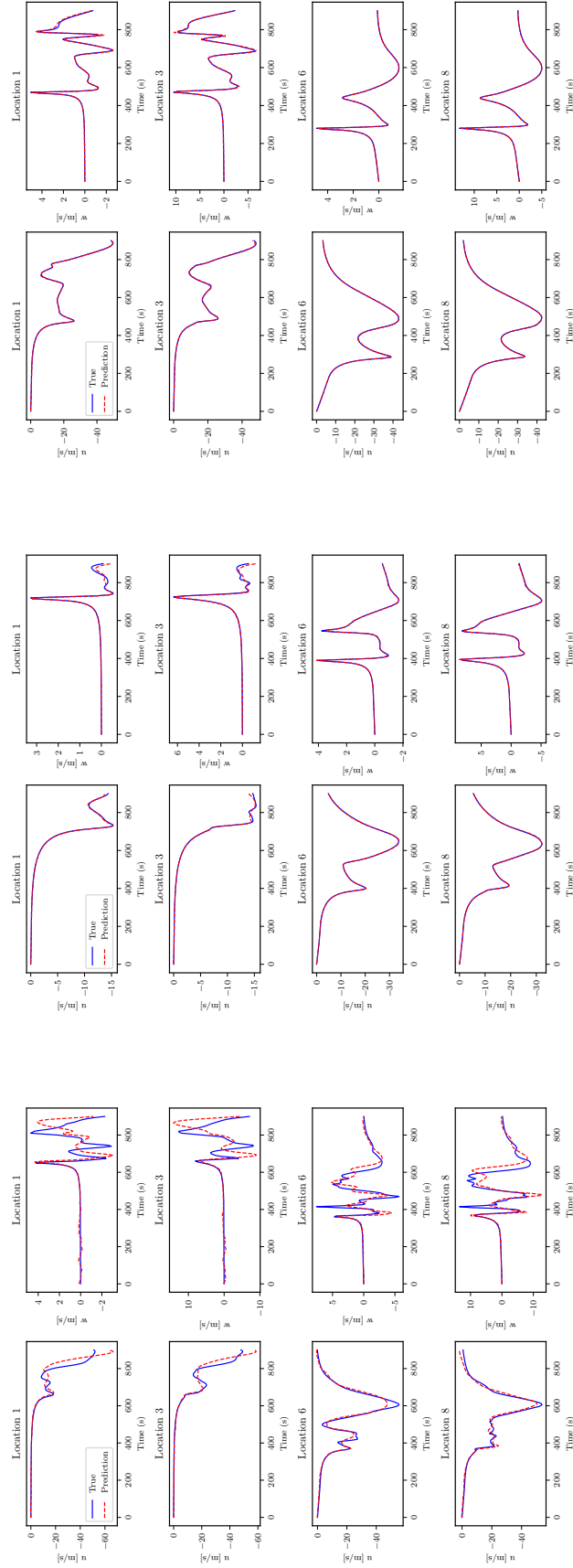



Figure A.13: ACB: As Figure A.12, for a sample with a stronger anomaly, reaching the state of three main vortices around $t = 600$ s instead of $t = 900$ s.

(a) Parameter estimates from the test sample with largest (worst) CRPS.

(b) Parameter estimates from the test sample with median CRPS.

Figure A.14: ACB, inverse problem: Samples drawn from the parameter distributions given continuous time series data. The true parameter values are marked with dashed lines, while the predicted distributions are given in blue shading.

32

(a) Sample with greatest (worst) error of FUSE.

(b) Sample with median error of FUSE.

(c) Sample with strong velocity gradients (Fig. A.13).

Figure A.15: ACB, forward problem: Velocity predictions with the FUSE model using the true parameters for different samples from the test set.

33

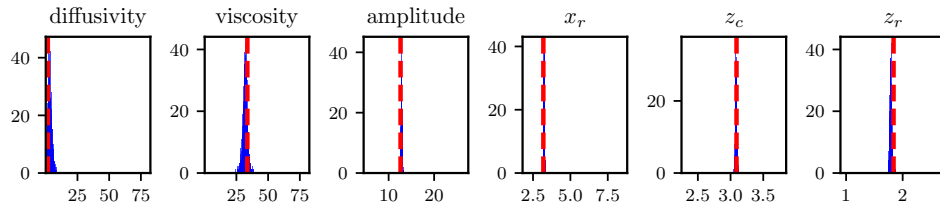
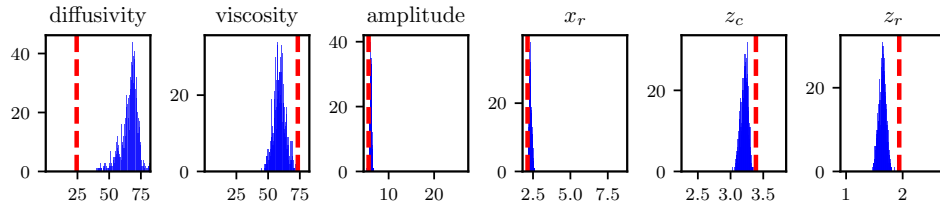
34

(a) Sample on which FUSE has the median CRPS.

(b) Sample on which FUSE has worst CRPS.

Figure A.16: ACB, inverse problem: Histograms of the parameters inferred from continuous time series measurements u, for different test samples. Given the very small amplitude and horizontal extend of the worst-case sample, this perturbation hardly reaches the sensor and the parameters are hence difficult to infer from the weak velocities measured.

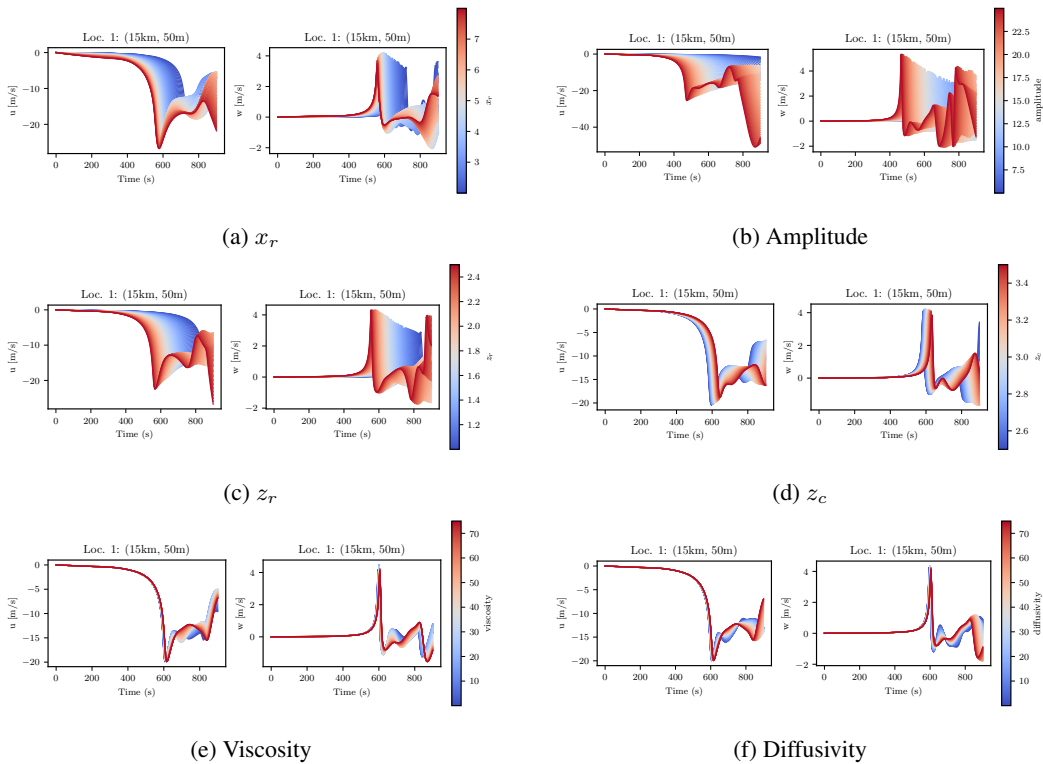



Figure A.17: ACB, sensitivity analysis: Fingerprints showing the model sensitivity to one parameter at a time at location 1, while keeping all others at their default value.

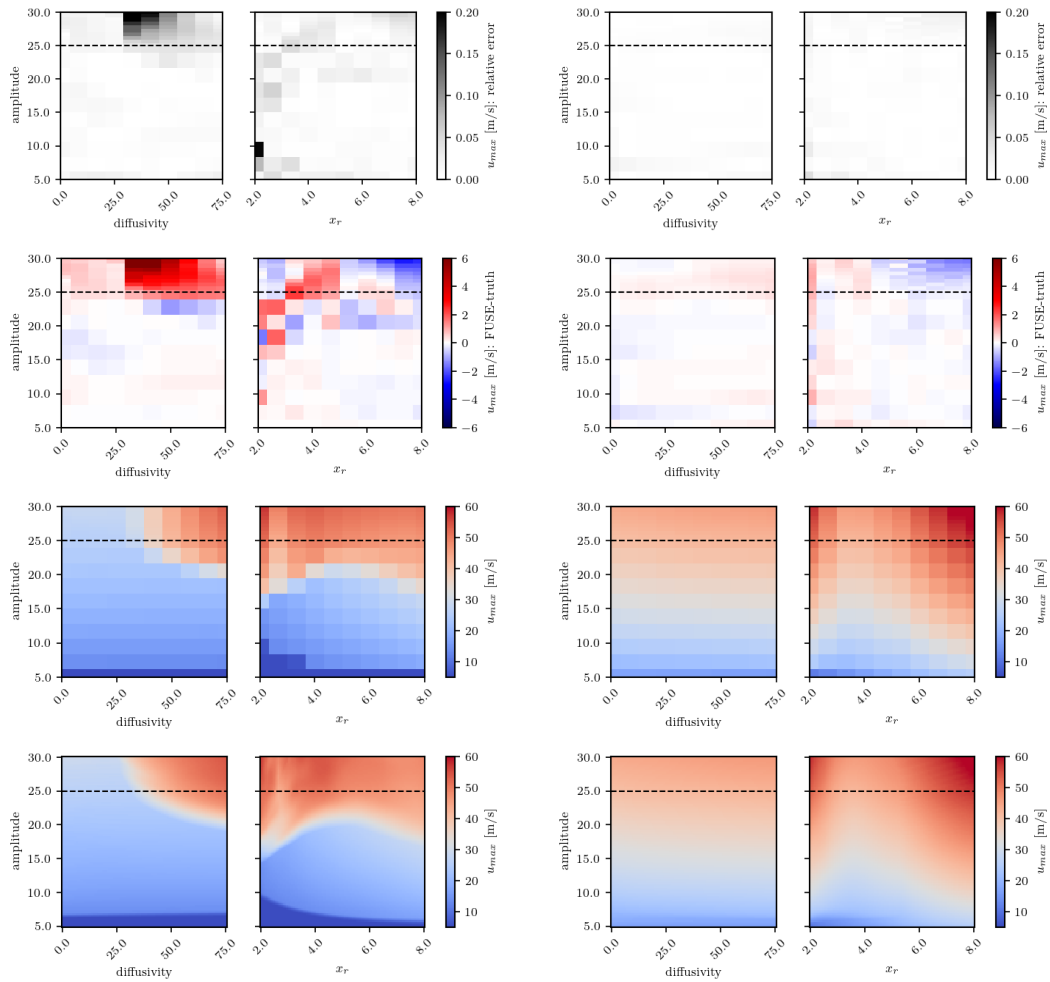
35

(a) Location 1 ($x = 15$ km, $z = 50$ m) (b) Location 5 ($x = 20$ km, $z = 50$ m)

Figure A.18: ACB, sensitivity analysis, continuation of Fig. 4: Validation of the FUSE model against numeric simulations on peak horizontal velocities u at location 1 (left, further away from the perturbation center) and 5 (right, closer). From top to bottom: relative error between FUSE and the numerical model, difference between FUSE and the numerical model, maximum velocity calculated by the numerical model, maximum velocity calculated by FUSE.

36

(a) FUSE model.   (b) Numerical ground truth.

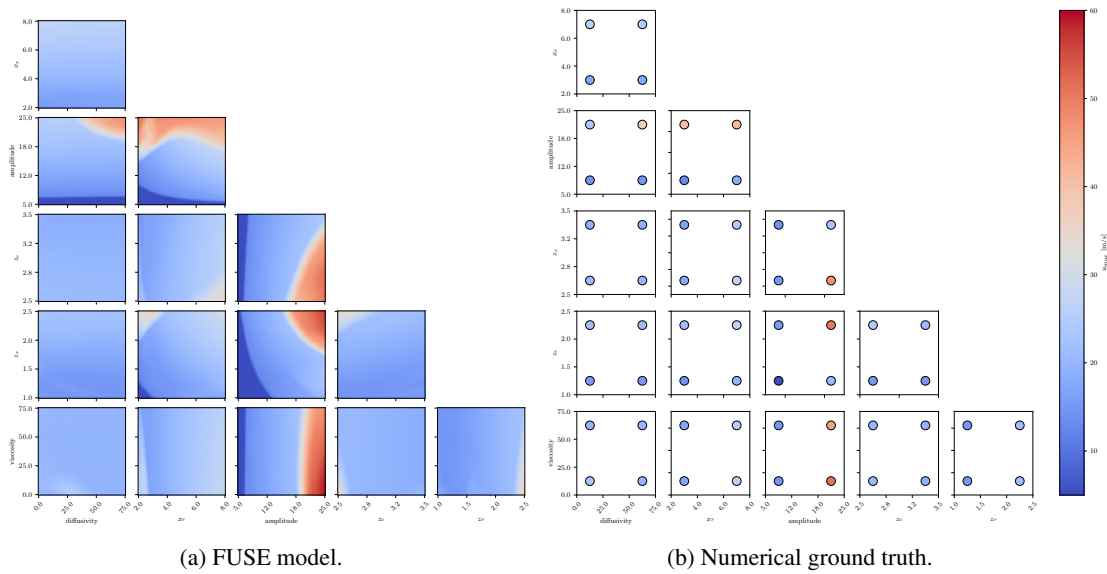



Figure A.19: ACB, sensitivity analysis: Peak horizontal velocities u at location 1 ($x = 15$ km, $z = 50$ m), sampled for pairwise combinations of the parameters, while keeping all others at their default value. For the neural model (a), 100 samples are drawn for each parameter pair, corresponding to 150,000 evaluations. For the numerical model (b), four simulations were run per pair of parameters, with each taking values corresponding to $1/6$ and $5/6$ of the parameter range, corresponding to 60 model evaluations.

(a) FUSE model.   (b) Numerical ground truth.

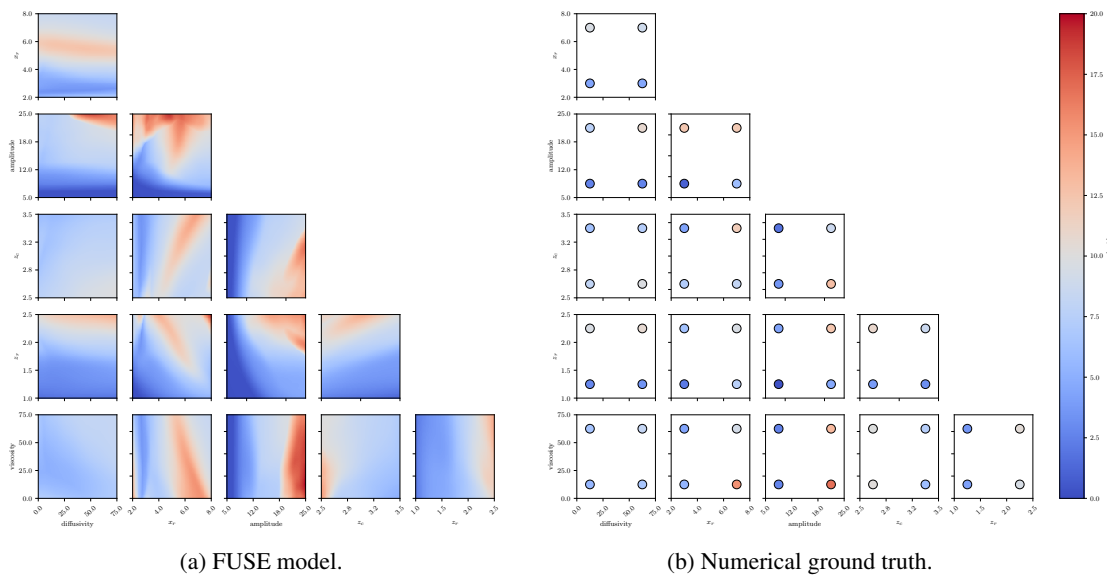



Figure A.20: ACB, forward problem: Same as Figure A.19, at location 5 ($x = 20$ km, $z = 2000$ m).

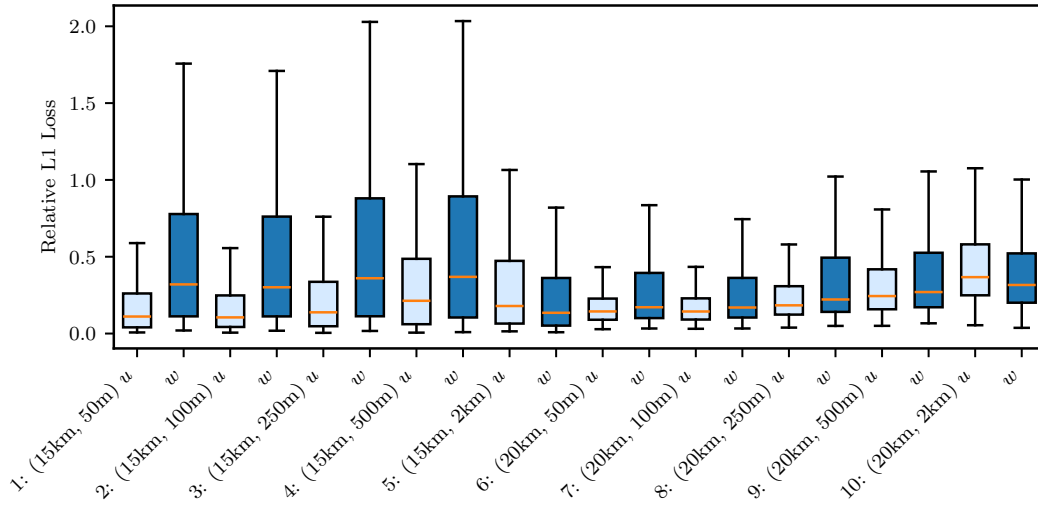
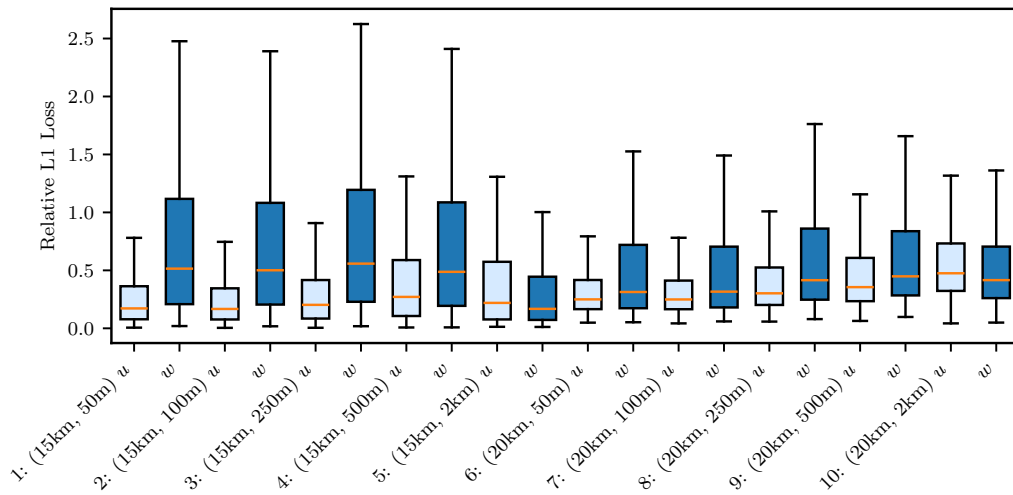
37

(a) Forward problem: Error between the prediction of s based on the true parameters ξ^* and the true output time series s.

(b) Unified prediction: Error between the predicted ensemble mean $\bar{s}$ based on the input time series u, and the true output time series s.

Figure A.21: ACB: Box plots of relative errors in the time series predictions at each location.

A.5 Properties of the Total Variation Metric

Consider the total variation distance $d(\mu, \mu^*)$, where μ and μ^* are measures over $u \in \mathcal{U}$, which is defined as

$$
\begin{aligned}
d(\mu, \mu^*) &= \sup_{A \subseteq \mathcal{U}} |\mu(A) - \mu^*(A)|, \\
&= \sup_{A \subseteq \mathcal{U}} |\int_{\mathcal{U}} \mu(u) du - \int_{\mathcal{U}} \mu^*(u) du|.
\end{aligned}
\tag{23}
$$

Further, consider a random vector $\xi \in \Xi \subseteq \mathbb{R}^m$, the function $h : \Xi \to \mathcal{U}$, and the probability measures ρ and ρ^* over Ξ. Since $u = h(\xi)$, the measures μ and μ^* can be considered as the pushforward measures induced by ρ through the function h, $\mu = h_{\#\rho}$ and $\mu^* = h_{\#\rho^*}$. We assume all measures admit densities and use the same notation for measures $\mu(A)$, $A \subseteq \mathcal{U}$, and their densities $\mu(u)$, $u \in \mathcal{U}$, since the distinction is clear from the arguments.

We consider the conditional probability measures $\rho(\xi|u), \rho^*(\xi|u)$. Then, μ and μ^* are given by:

$$
\mu(u) = \int_{\Xi} \rho(\xi)\rho(\xi|u)d\xi, \quad \mu^*(u) = \int_{\Xi} \rho^*(\xi)\rho^*(\xi|u)d\xi,
$$

and the measure of set A is:

$$
\mu(A) = \int_A \mu(u) du = \int_A \int_{\Xi} \rho(\xi)\rho(\xi|u)d\xi du
$$

$$
\mu^*(A) = \int_A \mu^*(u) du = \int_A \int_{\Xi} \rho^*(\xi)\rho^*(\xi|u)d\xi du
$$

Assume u is uniquely determined by ξ through the map $u = h(\xi)$ and that $\rho(\xi|u)$ is hence highly concentrated at $\xi = h^{-1}(u)$ and therefore:

$$
\rho(\xi|u) = \delta(\xi - h^{-1}(u))
$$
$$
\rho^*(\xi|u) = \delta(\xi - h^{-1}(u))
$$

where δ is the Dirac delta function and

$$
\mu(u) = \int_{\Xi} \rho(\xi)\delta(\xi - h^{-1}(u))d\xi = \rho(h^{-1}(u))
$$

$$
\mu^*(u) = \int_{\Xi} \rho^*(\xi)\delta(\xi - h^{-1}(u))d\xi = \rho^*(h^{-1}(u)).
$$

The definition of the total variation is then written as:

$$
\begin{aligned}
d(\mu, \mu^*) &= \sup_{A \subseteq \mathcal{U}} \left| \int_A \rho(h^{-1}(u)) du - \int_A \rho^*(h^{-1}(u)) du \right| \\
&= \sup_{A \subseteq \mathcal{U}} \left| \rho(h^{-1}(A)) - \rho^*(h^{-1}(A)) \right| \\
&= d(\rho, \rho^*).
\end{aligned}
\tag{24}
$$

Therefore due to the unique definition of u from ξ via h, the two definitions of the total variation are equivalent. The relation between the total variation of the original and the conditional measures is then given by:

$$
d(\mu, \mu^*) = d(\rho, \rho^*) \leq \int_{\mathcal{U}} d(\rho(\xi|u), \rho^*(\xi|u))d\mu^* = d(\rho(\xi|u), \rho^*(\xi|u)).
\tag{25}
$$

38

A.6 Detailed Derivation of the Forward Objective

We start with $d(\tilde{\mathcal{G}}^\theta_{\#\mu}, \tilde{\mathcal{G}}_{\#\mu})$ and our goal is to bound this metric by the supervised operator learning objective. For a function u, its corresponding images $\tilde{\mathcal{G}}(u)$ and $\tilde{\mathcal{G}}^\theta(u)$, and for any set $A \in \mathcal{B}(\mathcal{S})$, where $\mathcal{B}$ is a Borel σ-algebra over $\mathcal{S}$:

$$\tilde{\mathcal{G}}_{\#\mu}(A) = \int_\mathcal{S} w_A(s)d\tilde{\mathcal{G}}_{\#\mu}(s), \quad \tilde{\mathcal{G}}^\theta_{\#\mu}(A) = \int_\mathcal{S} w_A(s)d\tilde{\mathcal{G}}^\theta_{\#\mu}(s),$$

where $w_A(s)$ is the indicator function that is one if $s \in A$ and zero otherwise. Let $\mathcal{B}'(\mathcal{U})$ be the σ-algebra on $\mathcal{U}$ generated by the pre-images of sets in $\mathcal{B}(\mathcal{S})$ under $\tilde{\mathcal{G}}$. Then,

$$
\begin{aligned}
d(\tilde{\mathcal{G}}_{\#\mu}, \tilde{\mathcal{G}}^\theta_{\#\mu}) &= \sup_{A \in \mathcal{B}(\mathcal{S})} \left| \int_\mathcal{S} w_A(s)d\tilde{\mathcal{G}}^\theta_{\#\mu}(s) - \int_\mathcal{S} w_A(s)d\tilde{\mathcal{G}}^\theta_{\#\mu}(s) \right| \\
&= \sup_{\tilde{A} \in \mathcal{B}'(\mathcal{U})} \left| \int_\mathcal{U} w_{\tilde{A}}(\tilde{\mathcal{G}}(u))d\mu(u) - \int_\mathcal{U} w_{\tilde{A}}(\tilde{\mathcal{G}}^\theta(u))d\mu(u) \right| \\
&= \sup_{\tilde{A} \in \mathcal{B}'(\mathcal{U})} \left| \int_\mathcal{U} (w_{\tilde{A}}(\tilde{\mathcal{G}}(u)) - w_{\tilde{A}}(\tilde{\mathcal{G}}^\theta(u))d\mu(u) \right| \\
&\le \left| \int_\mathcal{U} (\tilde{\mathcal{G}}(u) - \tilde{\mathcal{G}}^\theta(u))d\mu(u) \right| \\
&\le \int_\mathcal{U} \|\tilde{\mathcal{G}}(u) - \tilde{\mathcal{G}}^\theta(u)\|_{L^1(Y)}d\mu(u).
\end{aligned}
\tag{26}
$$

Here, $\| \cdot \|_{L^1(Y)}$ is the L^1 norm over the space of function $\mathcal{S}$. We are interested in parametric PDEs where we have access to parameters $\xi \in \Xi \subset \mathbb{R}^m$, the PDE parameters, sampled from ρ, a probability measure on $\mathbb{R}^m$. Operator learning involves a map between continuous functions, so an intermediate step needs to be considered to first lift the parameters ξ to a continuous space before performing the action of the operator. For this purpose, we can, for example, consider h to be the band-limited lifting described in the main text, composed using a lifting $l : \mathbb{R}^m \to \mathcal{B}_\Omega$ and an inverse Fourier Transform $F^{-1} : \mathcal{B}_\Omega \to \mathcal{U}$. In this formulation, l maps the parameters to k to the Paley–Wiener space of band-limited functions in Fourier space, $\mathcal{B}_\Omega = \{l \in L^2(\mathbb{R}) : \text{supp } l \subset [-\Omega, \Omega]\}$, thus truncating the frequencies that can be used to represent the data. We consider $\mu = h_{\#\rho(\xi)}$, the pushforward of $\rho(\xi)$ under h, and use a change of variables:

$$\int_\mathcal{U} f(u)d\mu(u) = \int_\Xi f(h(\xi))d\rho(\xi),$$

for any integrable function f. We can apply this property to:

$$
\begin{aligned}
d(\tilde{\mathcal{G}}_{\#\mu}, \tilde{\mathcal{G}}^\theta_{\#\mu}) &= \sup_{\tilde{A} \in \mathcal{B}'(\mathcal{U})} \left| \int_\mathcal{U} (w_{\tilde{A}}(\tilde{\mathcal{G}}(u)) - w_{\tilde{A}}(\tilde{\mathcal{G}}^\theta(u))d\mu(u) \right| \\
&= \sup_{\tilde{A} \in \mathcal{B}'(\mathcal{U})} \left| \int_\Xi (w_{\tilde{A}}(\tilde{\mathcal{G}}(h(\xi))) - w_{\tilde{A}}(\tilde{\mathcal{G}}^\theta(h(\xi)))d\rho(\xi) \right|.
\end{aligned}
$$

We can define the operators $\mathcal{G} : \Xi \to \mathcal{S}$ and $\mathcal{G}^\theta : \Xi \to \mathcal{S}$ and write:

$$
\begin{aligned}
d(\tilde{\mathcal{G}}_{\#\mu}, \tilde{\mathcal{G}}^\theta_{\#\mu}) &= \sup_{\tilde{A} \in \mathcal{B}'(\mathcal{S})} \left| \int_\Xi (w_{\tilde{A}}(\mathcal{G}(\xi)) - w_{\tilde{A}}(\mathcal{G}^\theta(\xi)))d\rho(\xi) \right| \\
&\le \int_\Xi \|\mathcal{G}(\xi) - \mathcal{G}^\theta(\xi))\|_{L^1(Y)}d\rho(\xi)
\end{aligned}
$$

Finally, we arrive at

$$d(\tilde{\mathcal{G}}_{\#\mu}, \tilde{\mathcal{G}}^\theta_{\#\mu}) \le \int_\Xi \|\mathcal{G}(\xi) - \mathcal{G}^\theta(\xi))\|_{L^1(Y)}d\rho(\xi). \tag{27}$$

39

A.7 Detailed Derivation of the Inverse Objective

In the supervised operator learning problem, we have access to samples u of functions but we do not have access to the distribution that generates u. Assume that $\tilde{\mathcal{G}}^\theta$ is Lipschitz continuous:

$$\|\tilde{\mathcal{G}}^\theta(u) - \tilde{\mathcal{G}}^\theta(v)\|_{L^1(Y)} \leq C\|u - v\|_{L^1(X)}.$$

Assuming that μ^ϕ is an approximation of μ, we can relate these two measures by the total variation of their respective pushforward measures under the forward surrogate $\mathcal{G}^\theta$ using the continuity properties of the operator (see Appendix A.5),

$$d(\tilde{\mathcal{G}}^\theta_{\#\mu^\phi}, \tilde{\mathcal{G}}^\theta_{\#\mu}) \leq C\, d(\mu^\phi, \mu),$$

for some constant $C \in \mathbb{R}$, which we generally assume to be equal to one. This property holds because of the continuity of $\tilde{\mathcal{G}}^\theta$ and because the total variation is preserved under continuous transformations. We assume that the functions u are generated by solving a parametric PDE, where the parametric uncertainty over ξ is propagated onto u through the likelihood $\mu(u|\xi)$. Performing parameter estimation corresponds to sampling from a conditional distribution $\rho^\phi(\xi|u)$ approximating the posterior $\rho(\xi|u)$. To do so, we first bound the previous distance by the total variation between $\rho^\phi(\xi|u)$ and $\rho(\xi|u)$ over Ξ (see Appendix A.5):

$$d(\mu^\phi, \mu) \leq d(\rho^\phi(\xi|u), \rho(\xi|u)). \tag{28}$$

Combining the two inequalities, we get

$$d(\tilde{\mathcal{G}}^\theta_{\#\mu^\phi}, \tilde{\mathcal{G}}^\theta_{\#\mu}) \leq d(\rho^\phi(\xi|u), \rho(\xi|u)), \tag{29}$$

which means that we can minimize the right hand side of the inequality (29).

A.8 Formulation of the FMPE loss

For our implementation of Flow-Matching Posterior Estimation (FMPE), we closely follow [60]. In the main text (Eqn. 5), we formulated the FMPE objective as

$$d(\rho^\phi(\xi|u), \rho(\xi|u)),$$

for the total variation distance d on measures over the parameter space $\Xi \subseteq \mathbb{R}^m$. We will minimize this distance in order to approximate the underlying distribution, $\rho^\phi \approx \rho$, and sample M samples from $\rho^\phi(\xi|u)$ with the help of FMPE. We also defined the transform (6),

$$\hat{u} = T^{\phi_1, \phi_2}(u) = \hat{u} = \tilde{h} \circ \mathcal{K}^{\phi_1} \circ \mathcal{P}^{\phi_2}(u),$$

which transforms an infinite-dimensional input $u \in \mathcal{U}$ to a finite-dimensional conditional information $\hat{u} \in \mathbb{R}^k$ in Fourier space.

The FMPE method learns an invertible transform between the target distribution of parameters, conditional on the additional information, and a standard normal distribution. This transform is modelled as an ordinary differential equation with artificial time dimension $t \in [0, 1]$ and solution $\psi_{t,\hat{u}} : \mathbb{R}^m \to \mathbb{R}^m$ as

$$\frac{d}{dt}\psi_{t,\hat{u}}(\xi) = v_{t,\hat{u}}(\psi_{t,\hat{u}}(\xi)), \quad \psi_{0,\hat{u}}(\xi) = \xi_0, \tag{30}$$

where ξ_0 follows a standard normal distribution. Trajectories in the artificial time t can be obtained as $\xi_t = \psi_{t,\hat{u}}(\xi)$. This way, the target probability $\rho(\xi|\hat{u})$ is approximated as

$$\rho^\phi(\xi|\hat{u}) = (\psi_{1,\hat{u}})_{\#\rho_0}(\xi) = \rho_0(\xi)\exp\left(-\int_0^1 \operatorname{div} v_{t,\hat{u}}(\xi_t)\mathrm{d}t\right),$$

by solving the transport equation $\frac{d}{dt}\rho_t + \operatorname{div}(\rho_t v_{t,\hat{u}}) = 0$.

During training, $\psi_{0,\hat{u}_i}(\xi_i)$ is fit to resemble a standard normal on the training samples $\{\xi_i, \hat{u}_i\}_{i=1}^n$. During evaluation, a standard normal sample ξ_0 is drawn, and transformed to a sample that approximately follows $\rho(\xi|u)$ as

$$\xi' = \psi_{1,\hat{u}_i}(\xi_0) \sim \rho(\xi|u). \tag{31}$$

In order to fully describe $\psi_{t,\hat{u}}$, it hence suffices to learn the trajectory velocity function $v_{t,\hat{u}}$. Let $p_t(\xi|\xi_1)$ be a sample-conditional Gaussian probability path with vector field $l_t(\xi_t|\xi_1)$, and let time be distributed as $t \sim p(t)$. Then, [60] formulate the FMPE loss as

$$\mathcal{L}^{FMPE}(\phi) = \mathbb{E}_{t \sim p(t), \xi \sim \rho(\xi|u), \hat{u} \sim p(\hat{u}|\xi_1), \xi_t \sim p_t(\xi_t|\xi_1)}||v_{t,T^{\phi_1,\phi_2}(u)}^{\phi_0}(\xi_t) - l_t(\xi_t|\xi)||_2^2. \tag{32}$$

40

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in this paper are clearly highlighted in the abstract and introduction. The main contributions have been listed clearly in the introduction through bullet points. The results section is structured such that it justifies these claims.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Critical assumptions and practical limitations are presented in the Discussion section.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All assumptions and formulas are clearly stated, numbered, and cross-referenced in the paper with theoretical justification and detailed derivation provided in the appendix.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup is described extensively, with model architectures and training details outlined in the appendix. Furthermore, we provide all code, data, and models (including baselines) required to reproduce the experiments outlined in this paper.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The data and code is provided in the supplementary material, uploaded with the paper submission.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details are described for the all experiments, with more extensive training details specified in the appendix and Supplementary Material. Additionally, all the selected model configurations are provided in the code.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

41

Justification: Errors or scores are reported by the mean+/-standard deviation across all test samples. Furthermore, figures displaying box plots of these errors and scores are presented in the appendix. The quantities that these numbers and figures represent are clearly stated in the respective text and captions.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The amount of compute required for each of the experiments, as well as the total amount of compute hours for all experiments, is outlined in the training details reported in the appendix.

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: This work complies with the NeuIPS code of ethics, with special attention to the data and model documentation, access to research artifacts, and disclosure of essential elements for reproducibility.

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work is primarily focused on a broader strategy to solve problems in scientific computing, i.e. solve systems of differential equations. Direct paths to negative applications, such as deepfakes or generation of misinformation are not present. Additionally, while this work presents experiments on synthetic cardiovascular data, it does not meet the NeurIPS guidelines to immediately cause negative societal impact without a prior evaluation protocol. For health data specifically, detailed evaluation protocols issued by public health and safety organizations must be considered to meet in-silico, in-vitro, and in-vivo evaluation standards.

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The data used in these experiments has a very low risk for misuse. The dataset which is released with this work is based on fluid dynamics for turbulent systems, while the other dataset used in experiments is a publicly available, synthetic dataset of cardiovascular models.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The dataset for full-body haemodynamics is open source and appropriate cited and credited. The atmospheric turbulence dataset is produced by the authors, but based on an openly available code cited in the text. Baseline models used for comparison are properly cited in the text.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

42

Answer: [Yes]

Justification: The code that reproduces the results is documented and easy to read. The documentation is contained in the actual code.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

43