

NeurAM: nonlinear dimensionality reduction for uncertainty quantification through neural active manifolds

Andrea Zanoni ^{*} Gianluca Geraci [†] Matteo Salvador ^{*}
Alison L. Marsden ^{*‡§} Daniele E. Schiavazzi [¶]

Abstract

We present a new approach for nonlinear dimensionality reduction, specifically designed for computationally expensive mathematical models. We leverage autoencoders to discover a one-dimensional *neural active manifold* (NeurAM) capturing the model output variability, plus a simultaneously learnt surrogate model with inputs on this manifold. The proposed dimensionality reduction framework can then be applied to perform outer loop many-query tasks, like sensitivity analysis and uncertainty propagation. In particular, we prove, both theoretically under idealized conditions, and numerically in challenging test cases, how NeurAM can be used to obtain multifidelity sampling estimators with reduced variance by sampling the models on the discovered low-dimensional and shared manifold among models. Several numerical examples illustrate the main features of the proposed dimensionality reduction strategy, and highlight its advantages with respect to existing approaches in the literature.

Keywords. autoencoders, dimensionality reduction, multifidelity estimators, sensitivity analysis, surrogate modeling, uncertainty quantification.

1 Introduction

Mathematical modeling has become an indispensable tool to understand real-world phenomena in physics, engineering, and social sciences [26, 29]. Mathematical models depend on parameters that might be unknown and/or uncertain. As the computational models become more complex, quantification of predictive uncertainty plays an increasingly important role to fully assess the validity and accuracy of the results they provide. However, standard techniques for uncertainty quantification (UQ), such as Monte Carlo methods, require a large number of samples to reliably estimate the statistical moments (e.g., mean and variance) of the stochastic response, and reaching high precision can be unfeasible if the model is computationally expensive. Moreover, if the number of input parameters of the model increases, most methods suffer from the curse of dimensionality, meaning that the complexity of the estimation task grows exponentially with the dimensionality, making the

^{*}Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA.

[†]Center for Computing Research, Sandia National Laboratories, Albuquerque, NM, USA.

[‡]Pediatric Cardiology, Stanford University, Stanford, CA, USA.

[§]Bioengineering, Stanford University, Stanford, CA, USA.

[¶]Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN, USA.

uncertainty quantification problem intractable [18]. In uncertainty quantification, as in any other many-query workflows such as optimization, global sensitivity analysis, and design, dimensionality reduction techniques become crucial. Dimensionality reduction offers a powerful approach to alleviate the computational cost of workflows that need to query the model numerous times, e.g., uncertainty quantification, by reducing the number of model parameters while preserving the main output features. Moreover, some mathematical models can be overparameterized, and reducing the dimensionality can help to capture the most relevant information, while disregarding redundant variables or unimportant combinations thereof.

Overviews of dimensionality reduction strategies can be found in, e.g., [8, 21]. We mention, in particular, two recent approaches related to sliced inverse regression [14, 25] that motivated our work. Active subspaces (AS) was first proposed to determine linear low-dimensional subspaces of maximum model variance in [35], and then thoroughly analyzed in [10, 12]. The *active* directions of a model consist of the eigenvectors corresponding to the largest eigenvalues from the expected gradient covariance matrix. AS have been demonstrated in numerous contexts and applications including Bayesian inverse problems [13], sensitivity analysis [11], and multifidelity dimensionality reduction [23]. The main limitation of AS, inherited from its linear character, is that it only captures the model variation on average, and, therefore, the existence of a lower dimensional manifold cannot be guaranteed [7]. A second approach called active manifolds (AM) [7] overcomes some of the limitations of AS. This approach is based on a local construction of level sets which approximates any continuously differentiable multi-dimensional function using a surrogate defined over a one-dimensional manifold, obtained through a non linear transformation. The main advantages of AM over AS are that they always provide a one-dimensional reduced manifold, give a more accurate surrogate, and enable more informative sensitivity analysis. Unfortunately, as discussed in [7], the geometric nature of the AM algorithm requires that the function’s level sets are fully contained in the original domain of the function. Indeed, the hypothesis of AM is that any arbitrary multi-dimensional \mathcal{C}^1 function can be related to a one-dimensional \mathcal{C}^1 counterpart where each point of the function in the original space is mapped to a point on the one-dimensional AM moving tangentially to the corresponding level set. As a consequence, if a point lies on a level set that leaves the domain, it is impossible to map it to the AM. Additionally, both AS and AM methods require the computation of the model output gradient, which is often unavailable for complex scientific codes, or computationally expensive to approximate for high-dimensional problems. We note that alternative approaches like the so-called Adaptive Basis [40, 45] are able to overcome the need for gradients by relying on the construction of a first order polynomial chaos surrogate [24]. However, since this approach is still limited to the identification of a linear manifold, in the rest of the paper we only consider AS as representative of both methods.

In this work we propose a novel methodology inspired by AM, that does not rely on the gradient of the model, and discovers a transformation from the original input space to a one-dimensional nonlinear manifold without being restricted by level sets. We call this methodology *neural active manifolds* (NeurAM). Our approach is based on autoencoders [2], a well established approach for unsupervised dimensionality reduction with neural networks. The architecture consists of an encoder, which maps the input into a lower-dimensional latent space, and a decoder, which maps points from the latent space back to the original input space. Classic autoencoders for supervised tasks have been used in [5] to determine active manifolds that are then employed to accelerate multidisciplinary analysis and optimization. This approach relies on a snapshot matrix built using an auxiliary cheaper

model. However, unlike unsupervised learning, where autoencoders are used to minimize the reconstruction error, we are interested in determining a one-dimensional manifold in the space of parameters capturing the entire variability of the model output. To do so, we design a computational pipeline where a surrogate model with inputs on a one-dimensional latent space is leveraged by the autoencoder to identify the manifold on which the representation of the function can be obtained with minimal error. This surrogate, which is parameterized as a dense neural network, is trained at the same time as the autoencoder’s parameters, defining the encoder and decoder.

The resulting one-dimensional surrogate computed by NeurAM can then be easily employed for many-query tasks, like sensitivity analysis and uncertainty propagation. In particular, we first show how to perform sensitivity analysis along the manifold and derive both local and global indices for the input parameters with respect to a scalar output. Our indices, in contrast to more classical indices, such as Sobol’ [38] and Borgonovo [32], are defined along a one-dimensional manifold.

Then, we focus on the problem of uncertainty propagation, which can easily become unfeasible for computationally expensive models. In order to mitigate this issue, multifidelity estimators [4, 19, 28, 30] rely on cheaper-to-evaluate, but correlated, low-fidelity models to achieve variance reduction. The statistical correlation among models is directly related to the attainable variance reduction, which corresponds to a greater estimator precision when compared to a single fidelity Monte Carlo estimator with the same computational cost. Therefore, for poorly correlated model pairs such as those characterized by a dissimilar number of input parameters, methodologies to increase the correlation between the fidelities have been recently proposed, and are based on approaches for linear dimensionality reduction such as AS [16] and adaptive basis [43]. A first step towards the extension to nonlinear dimensionality reduction using autoencoders is discussed in [41, 42]. Following this line of work, we leverage NeurAM to enhance the performance of sampling multifidelity estimators, complementing numerical test cases showing variance reduction of the resulting estimators with a theoretical analysis under simplified assumptions. In particular, we demonstrate that, under such assumptions, NeurAM provides new sampling locations, in the original models’ inputs, that correspond to shared samples in the low-dimensional manifold leading to a bi-fidelity correlation that is never smaller than the one produced by the original samples. Importantly, we note that the approach proposed here could be directly leveraged in the context of the multifidelity construction of surrogates, e.g., following recent work discussed in [44]. For simplicity of exposure, in this manuscript, we limit the use of NeurAM to the case of sampling estimators and we leave its integration with surrogate techniques to a future study.

The main contributions of this work include:

- introducing an algorithm to learn a one-dimensional active manifold aided by a simultaneously trained surrogate model;
- proposing a way to leverage NeurAM for sensitivity analysis;
- showing how to use nonlinear dimensionality reduction to improve the correlation between low- and high-fidelity models, and consequently to obtain multifidelity sampling estimators of reduced variance;
- providing a theoretical result under specific assumptions, showing that the correlation obtained from NeurAM is never smaller than the initial correlation.

The rest of the paper is organized as follows. In Section 2 we introduce NeurAM, and present

possible applications of our method in the field of uncertainty quantification in Section 3. We then discuss sensitivity analysis in Section 3.1 and multifidelity uncertainty propagation in Section 3.2. Then, in Section 4 we demonstrate the proposed approach on several numerical examples of increasing complexity. Finally, conclusions and possible future research directions are presented in Section 5.

2 Methodology

In this section we present our methodology to reduce the dimensionality of a computationally expensive model $\mathcal{Q}: \mathbb{R}^d \rightarrow \mathbb{R}$ with $d > 1$, and derive a one-dimensional surrogate. Let us assume that we are given a set of N realizations $\{(x_n, \mathcal{Q}(x_n))\}_{n=1}^N$, where the samples $\{x_n\}_{n=1}^N$ are drawn from an input distribution μ . Using the notation of the autoencoders, consider an encoder $\mathcal{E}: \mathbb{R}^d \rightarrow \mathbb{R}$ to reduce the dimensionality, and a decoder $\mathcal{D}: \mathbb{R} \rightarrow \mathbb{R}^d$ to re-map a given sample to the original input space. A fundamental property of NeurAM is that it aims to represent the entire variability of the model \mathcal{Q} , therefore the encoder and the decoder need to satisfy $\mathcal{Q}(x) \simeq \mathcal{Q}(\mathcal{D}(\mathcal{E}(x)))$; see also [41]. This implies that \mathcal{E} and \mathcal{D} can be obtained by minimizing the loss function

$$\ell(\mathcal{E}, \mathcal{D}) = \mathbb{E}^\mu \left[(\mathcal{Q}(X) - \mathcal{Q}(\mathcal{D}(\mathcal{E}(X))))^2 \right], \quad (2.1)$$

where the superscript denotes the fact that the expectation is computed with respect to the measure μ , i.e., $X \sim \mu$. In order to solve the minimization problem, the encoder and the decoder need to be parameterized as $\mathcal{E}(\cdot; \alpha)$ and $\mathcal{D}(\cdot; \beta)$, and the expectation is approximated by its Monte Carlo approximation, yielding the discrete optimization problem

$$\tilde{\ell}(\alpha, \beta) = \frac{1}{N} \sum_{n=1}^N (\mathcal{Q}(x_n) - \mathcal{Q}(\mathcal{D}(\mathcal{E}(x_n; \alpha); \beta)))^2. \quad (2.2)$$

In practice, solving (2.2) without incurring additional model evaluations is not feasible. In fact, any optimization algorithm, e.g., (stochastic) gradient descent, would require a *new* evaluation of the model at $\mathcal{D}(\mathcal{E}(x_n; \alpha); \beta)$, for $n = 1, \dots, N$, at each iteration.

To overcome this limitation, we propose to train a surrogate model $\mathcal{S}: \mathbb{R} \rightarrow \mathbb{R}$ on the latent space, and learn it simultaneously with the nonlinear transformation from the original points in \mathbb{R}^d to the corresponding points in \mathbb{R} on the neural active manifold. We generalize the loss function in equation (2.1) to account for the surrogate contribution

$$\begin{aligned} \mathcal{L}(\mathcal{E}, \mathcal{D}, \mathcal{S}) = & \mathbb{E}^\mu \left[(\mathcal{Q}(X) - \mathcal{S}(\mathcal{E}(\mathcal{D}(\mathcal{E}(X))))))^2 \right] \\ & + \mathbb{E}^\mu \left[(\mathcal{Q}(X) - \mathcal{S}(\mathcal{E}(X)))^2 \right] \\ & + \mathbb{E}^\mu \left[(\mathcal{D}(\mathcal{E}(X)) - \mathcal{D}(\mathcal{E}(\mathcal{D}(\mathcal{E}(X)))))^2 \right], \end{aligned} \quad (2.3)$$

where the first term corresponds to the one in the original loss ℓ with the surrogate \mathcal{S} defined on the latent space replacing the full model, the second term refers to the error of the surrogate model, and the third term enforces the fact that a point on the NeurAM (\tilde{x}) must be mapped in the point itself ($\tilde{\tilde{x}}$), i.e., the second part of the network in Figure 1, where a schematic representation of the approach is provided, acts as a classical autoencoder. In particular, we remark that we do not want to necessarily reconstruct all the original inputs, i.e., $\mathcal{D}(\mathcal{E}(x)) \neq x$ in general.

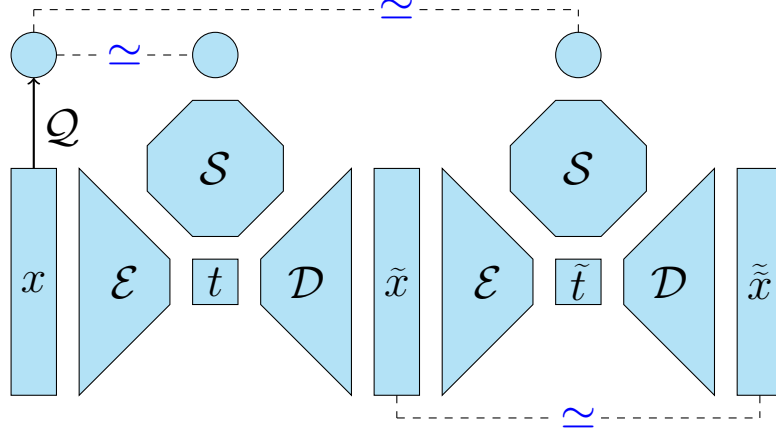


Figure 1: Schematic representation of the NeurAM architecture, where the symbols \simeq are used to indicate the three terms in the loss function in equation (2.3).

We notice that the NeurAM is parameterized by the decoder $\mathcal{D}(t)$ for all t in the interval

$$\mathcal{T} = \left[\min_{x \in \mathbb{R}^d} \mathcal{E}(x), \max_{x \in \mathbb{R}^d} \mathcal{E}(x) \right], \quad (2.4)$$

and the one-dimensional analogue of the model \mathcal{Q} on the manifold is given by \mathcal{S} . Moreover, this methodology provides a surrogate model \mathcal{Q}_S for \mathcal{Q} defined as

$$\mathcal{Q}_S(x) = \mathcal{S}(\mathcal{E}(x)). \quad (2.5)$$

The following result shows that the minimization problem with loss function (2.3) admits a solution.

Proposition 2.1. *The loss function \mathcal{L} defined in (2.3) has at least one global minimizer.*

Proof. First, notice that $\mathcal{L} \geq 0$. Then, define

$$\mathcal{S}^* = \mathcal{I} \quad \text{and} \quad \mathcal{E}^* = \mathcal{Q},$$

and set \mathcal{D}^* to be a right inverse of \mathcal{Q} such that

$$\mathcal{Q} \circ \mathcal{D}^* = \mathcal{I}.$$

We remark that the right inverse exists if and only if the model \mathcal{Q} is surjective, but it is always possible to restrict the codomain of \mathcal{Q} in order to make it surjective, since it maps a d -dimensional domain into a one-dimensional one. We now notice that $\mathcal{L}(\mathcal{E}^*, \mathcal{D}^*, \mathcal{S}^*) = 0$ independently of the measure μ , and therefore $(\mathcal{E}^*, \mathcal{D}^*, \mathcal{S}^*)$ is a global minimizer of \mathcal{L} . \square

Remark 2.2. Proposition 2.1 gives the existence of a global minimizer of \mathcal{L} , but does not guarantee its uniqueness. In particular, it is possible to show that \mathcal{L} has multiple global minimizers. First, from the proof of Proposition 2.1 notice that the choice of the right inverse \mathcal{D}^* is not unique, since \mathcal{Q} is not injective. Moreover, a family of minimizers could be obtained by setting

$$\mathcal{S}^* = \frac{1}{a} \mathcal{I}, \quad \mathcal{E}^* = a \mathcal{Q}, \quad (a \mathcal{Q}) \circ \mathcal{D}^* = \mathcal{I},$$

for a parameter $a \in \mathbb{R}$, $a \neq 0$.

The solution provided by the proof of Proposition 2.1 is not useful from a practical point of view, as it relies on the model \mathcal{Q} , which is computationally expensive, and requires the computation of its right inverse. Therefore, we consider the parameterized encoder $\mathcal{E}(\cdot; \alpha)$, decoder $\mathcal{D}(\cdot; \beta)$, and surrogate model $\mathcal{S}(\cdot; \gamma)$, which are represented by neural networks, and compute their optimal parameters by minimizing the approximated loss function

$$\begin{aligned}\tilde{\mathcal{L}}(\alpha, \beta, \gamma) = & \frac{1}{N} \sum_{n=1}^N (\mathcal{Q}(x_n) - \mathcal{S}(\mathcal{E}(\mathcal{D}(\mathcal{E}(x_n; \alpha); \beta); \alpha); \gamma))^2 \\ & + \frac{1}{N} \sum_{n=1}^N (\mathcal{Q}(x_n) - \mathcal{S}(\mathcal{E}(x_n; \alpha); \gamma))^2 \\ & + \frac{1}{N} \sum_{n=1}^N (\mathcal{D}(\mathcal{E}(x_n; \alpha); \beta) - \mathcal{D}(\mathcal{E}(\mathcal{D}(\mathcal{E}(x_n; \alpha); \beta); \alpha); \beta))^2.\end{aligned}\tag{2.6}$$

In Algorithm 1 we summarize the main steps to construct a neural active manifold. The closeness of the solution of the optimization problem to the global minimizer given in Proposition 2.1 is dependent on the complexity of the model \mathcal{Q} , the expressive power of encoder and decoder, and the number N of samples. Nevertheless, even if we cannot achieve a loss function equal to zero in practice, in Section 4 we show that the NeurAM determined by this procedure provides a valid representation of the model \mathcal{Q} in one dimension.

Algorithm 1: NeurAM

Input: Model \mathcal{Q}

Input distribution μ

Number of samples N

Output: Neural active manifold $\mathcal{D}(t; \beta^*)$ for t in \mathcal{T}

Encoder $\mathcal{E}(\cdot; \alpha^*)$ to obtain the latent space

Surrogate model $\mathcal{S}(\cdot; \gamma^*)$ on the latent space

Surrogate model \mathcal{Q}_S of \mathcal{Q}

- 1: Parameterize the encoder through a neural network $\mathcal{E}(\cdot; \alpha)$.
 - 2: Parameterize the decoder through a neural network $\mathcal{D}(\cdot; \beta)$.
 - 3: Parameterize the surrogate model through a neural network $\mathcal{S}(\cdot; \gamma)$.
 - 4: Draw the samples and evaluate the model to get $\{(x_n, \mathcal{Q}(x_n))\}_{n=1}^N$ with $x_n \sim \mu$.
 - 5: Define the loss function $\tilde{\mathcal{L}}(\alpha, \beta, \gamma)$ given in equation (2.6).
 - 6: Compute the optimal parameters $(\alpha^*, \beta^*, \gamma^*) = \arg \min \tilde{\mathcal{L}}(\alpha, \beta, \gamma)$.
 - 7: Compute the domain of the neural active manifold \mathcal{T} from equation (2.4).
 - 8: Define the surrogate model \mathcal{Q}_S using equation (2.5).
-

Remark 2.3. Even if learning the NeurAM can be computationally expensive depending on the complexity of the neural networks representing encoder $\mathcal{E}(\cdot; \alpha)$, decoder $\mathcal{D}(\cdot; \beta)$, and surrogate model $\mathcal{S}(\cdot; \gamma)$, the main advantages with respect to other techniques in the literature, such as active subspaces [12] or active manifolds [7], are that the evaluation of the gradient of the model $\nabla \mathcal{Q}$ is never required and that we can identify a nonlinear

manifold described by an arbitrary transformation from the original space, regardless of the function level sets. Moreover, we also train a surrogate model \mathcal{Q}_S without any additional cost.

Remark 2.4. If the model \mathcal{Q} is not defined on the whole space \mathbb{R}^d , but only on a domain $\Omega \subset \mathbb{R}^d$, then we need to enforce the NeurAM to lie on Ω . Hence, we need to impose the codomain of the decoder \mathcal{D} to be included in Ω . This constraint can be easily enforced from the practical viewpoint if the domain is a box of the form $\Omega = \prod_{i=1}^d [a_i, b_i]$ with $a_i < b_i$ for all $i = 1, \dots, d$, as it is usually the case in applications. Otherwise, if the input parameters are correlated, one would need an invertible map h from the domain Ω to a simpler box or to the whole space to decorrelate the variables. We leave the problem of determining such a map, which could be done using, e.g., normalizing flows [20], for future work.

3 Application to uncertainty quantification

In this section we discuss possible applications of NeurAM in the field of uncertainty quantification. In addition to providing a surrogate model, our methodology can be used for sensitivity analysis. Moreover, the one-dimensional latent space can be employed to enhance the performance of multifidelity estimators in the context of uncertainty propagation.

3.1 Sensitivity analysis

Similar to AM [7], NeurAM can be used to perform a more informative sensitivity analysis than AS [12]. By computing the derivatives of the surrogate model \mathcal{Q}_S with respect to the input parameters along the NeurAM, we can perform local sensitivity analysis, and quantify how the relevance, and consequently the identifiability, of each parameter varies along the manifold, hence providing a dynamic ranking. In particular, consider the quantities λ_i for all $i = 1, \dots, d$ defined as

$$\lambda_i(x) = \frac{1}{\|\nabla \mathcal{Q}_S(x)\|^2} \left| \frac{\partial \mathcal{Q}_S}{\partial x_i}(x) \right|^2,$$

and notice that they represent how much a single input parameter contributes to the variation of the model. Therefore, by evaluating the quantities λ_i along the NeurAM, we obtain a measure θ_i of importance of each parameter that varies along the manifold, and which is given by

$$\theta_i(t) = \frac{1}{\|\nabla \mathcal{Q}_S(\mathcal{D}(t))\|^2} \left| \frac{\partial \mathcal{Q}_S}{\partial x_i}(\mathcal{D}(t)) \right|^2, \quad t \in \mathcal{T}, \quad (3.1)$$

where, by definition (2.5) and using the chain rule, we have

$$\frac{\partial \mathcal{Q}_S}{\partial x_i}(\mathcal{D}(t)) = \mathcal{S}'(\mathcal{E}(\mathcal{D}(t))) \frac{\partial \mathcal{E}}{\partial x_i}(\mathcal{D}(t)),$$

with \mathcal{S}' being the derivative of the surrogate \mathcal{S} with respect to the latent variable t . Note that, from the practical point of view, the derivatives can be computed using automatic differentiation. We also remark that the local indices sum up to one at all points in the manifold, i.e., for all $t \in \mathcal{T}$

$$\sum_{i=1}^d \theta_i(t) = 1. \quad (3.2)$$

Moreover, letting Γ be the NeurAM, we can define global indices by integrating the dynamic indices along the manifold as

$$\Theta_i = \frac{1}{|\Gamma|} \int_{\Gamma} \lambda_i d\Gamma,$$

for all $i = 1, \dots, d$, which give an overall ranking of the importance of the input parameters. The line integrals can then be computed from the parameterization \mathcal{D} of the manifold Γ yielding the indices

$$\Theta_i = \frac{\int_{\mathcal{T}} \theta_i(t) |\mathcal{D}'(t)| dt}{\int_{\mathcal{T}} |\mathcal{D}'(t)| dt}, \quad (3.3)$$

where \mathcal{D}' stands for the derivative of the decoder \mathcal{D} with respect to the latent variable t , and which, due to the equality (3.2), satisfy

$$\sum_{i=1}^d \Theta_i = 1.$$

These indices, depending on whether the input parameters are normalized or not, might be sensitive to units. In fact, even if the variable $t \in \mathcal{T}$ describing the manifold is common for all the parameters, the components of the output of the decoder \mathcal{D} have the same units as the inputs parameters of the model \mathcal{Q} . It is therefore always important to first normalize the dataset. We note that in this novel approach for sensitivity analysis based on a one-dimensional reduction, both the local and global indices give a ranking for the relevance of the input parameters without taking into account their interactions. Nevertheless, if one is interested in extracting more information regarding the interaction between variables, classic sensitivity indices, e.g., Sobol', can also be computed at a minimal computational cost by leveraging the surrogate model \mathcal{Q}_S provided by the NeurAM algorithm.

We finally remark that, even if we suggest to use the surrogate model \mathcal{Q}_S for sensitivity analysis, we do not always expect \mathcal{Q}_S to be a perfect approximation of the original model \mathcal{Q} . This surrogate is indeed built with the main purpose of helping to determine the NeurAM. Nevertheless, even if the error between \mathcal{Q} and \mathcal{Q}_S is not negligible, the surrogate model can still be useful for discovering an active manifold, and identifying the most relevant input parameters through sensitivity analysis.

3.2 Multifidelity uncertainty propagation

Inspired by the works [41, 42], neural active manifolds can be used to increase the correlation between high-fidelity and low-fidelity models in the framework of multifidelity uncertainty propagation. For sake of generality, we limit our analysis to multifidelity Monte Carlo estimators [28] with a single low-fidelity model. Nevertheless, we remark that a straightforward extension to other estimators, e.g., Approximate Control Variate [19] and multiple low-fidelity sources, is also possible. We assume a computationally expensive high-fidelity model $\mathcal{Q}_{\text{HF}}: \mathbb{R}^d \rightarrow \mathbb{R}$ and a cheaper low-fidelity model $\mathcal{Q}_{\text{LF}}: \mathbb{R}^d \rightarrow \mathbb{R}$, and consider the problem of estimating the quantity

$$q = \mathbb{E}^{\mu}[\mathcal{Q}_{\text{HF}}(X)],$$

for some probability distribution μ on \mathbb{R}^d . We want to employ the multifidelity Monte Carlo estimator introduced in [28]. Let $w = \mathcal{C}_{\text{HF}}/\mathcal{C}_{\text{LF}}$ be the cost ratio between the two fidelities, and let \mathcal{B} be the available computational budget given in terms of evaluations of the high-fidelity model, i.e.,

$$\mathcal{B} = N_{\text{HF}} + wN_{\text{LF}},$$

where N_{HF} and N_{LF} are the number of high-fidelity and low-fidelity evaluations, respectively. By solving an optimal allocation problem, it is possible to split the total computational budget between the high-fidelity and the low-fidelity models in such a way that the variance of the resulting multifidelity Monte Carlo estimator is minimized [30]. In particular, the solution of the allocation problem is given by

$$N_{\text{HF}} = \frac{\mathcal{B}}{1 + w\gamma} \quad \text{and} \quad N_{\text{LF}} = \gamma N_{\text{HF}} = \frac{\gamma \mathcal{B}}{1 + w\gamma}, \quad \text{with} \quad \gamma = \sqrt{\frac{\rho^2}{w(1 - \rho^2)}},$$

where ρ is the Pearson correlation coefficient between the high-fidelity and low-fidelity models

$$\rho = \frac{\mathbb{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \mathcal{Q}_{\text{LF}}(X))}{\sqrt{\mathbb{Var}^\mu[\mathcal{Q}_{\text{HF}}(X)] \mathbb{Var}^\mu[\mathcal{Q}_{\text{LF}}(X)]}}. \quad (3.4)$$

After fixing the number of evaluations N_{HF} and N_{LF} , we are now ready to define the multifidelity Monte Carlo estimator

$$\hat{q} = \frac{1}{N_{\text{HF}}} \sum_{n=1}^{N_{\text{HF}}} \mathcal{Q}_{\text{HF}}(x_n) - \beta \left(\frac{1}{N_{\text{HF}}} \sum_{n=1}^{N_{\text{HF}}} \mathcal{Q}_{\text{LF}}(x_n) - \frac{1}{N_{\text{LF}}} \sum_{n=1}^{N_{\text{LF}}} \mathcal{Q}_{\text{LF}}(x_n) \right), \quad (3.5)$$

where the coefficient β is given by

$$\beta = \frac{\mathbb{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \mathcal{Q}_{\text{LF}}(X))}{\mathbb{Var}^\mu[\mathcal{Q}_{\text{LF}}(X)]}, \quad (3.6)$$

and the samples $\{x_n\}_{n=1}^{N_{\text{LF}}}$ are drawn from the distribution μ . We remark that the estimator \hat{q} is unbiased, i.e., $\mathbb{E}^\mu[\hat{q}] = q$, and that the correlation ρ in (3.4) and the coefficient β in (3.6) can be estimated from a pilot sample $\{x_n\}_{n=1}^N$ with $N < N_{\text{LF}}$. The performance of the multifidelity Monte Carlo estimator, i.e., its variance, is strongly dependent on the correlation ρ . In particular, we have

$$\mathbb{Var}^\mu[\hat{q}] = \frac{1}{\mathcal{B}} \mathbb{Var}^\mu[\mathcal{Q}_{\text{HF}}(X)] \left(\sqrt{1 - \rho^2} + \sqrt{w\rho^2} \right)^2,$$

which is a decreasing function in ρ^2 as long as $\rho^2 > w/(1 + w)$. Therefore, for a fixed computational budget, the larger the correlation is in modulus, the smaller the variance is, and consequently the estimation is more precise.

We now adopt the methodology introduced in [41] based on nonlinear dimensionality reduction to increase the correlation between the fidelities, and therefore decrease the variance of the estimator. The main limitation in [41] is the need for a surrogate model to discover the lower-dimensional manifold. We overcome this restriction by using neural active manifolds, which do not rely on any surrogate model, but build a surrogate model on the latent space simultaneously with the learning of the lower-dimensional manifold. Moreover, in [41] a shared space between the high-fidelity and low-fidelity models is built employing normalizing flows, which, however, may suffer from poor performance in presence of limited available data. Since our NeurAM is one-dimensional, we replace normalizing flows by the inverse transform method, and consider the uniform distribution $\mathcal{U}([0, 1])$ as the reference distribution in the shared space.

We proceed as follows. We first compute the NeurAM for both the high-fidelity and low-fidelity models by minimizing the loss function (2.6), and thus obtaining

$$(\mathcal{E}_{\text{HF}}(\cdot; \alpha_{\text{HF}}), \mathcal{D}_{\text{HF}}(\cdot; \beta_{\text{HF}}), \mathcal{S}_{\text{HF}}(\cdot; \gamma_{\text{HF}})) \quad \text{and} \quad (\mathcal{E}_{\text{LF}}(\cdot; \alpha_{\text{LF}}), \mathcal{D}_{\text{LF}}(\cdot; \beta_{\text{LF}}), \mathcal{S}_{\text{LF}}(\cdot; \gamma_{\text{LF}})).$$

Then, let \mathcal{F}_{HF} and \mathcal{F}_{LF} be the cumulative distribution functions in the latent spaces of the two fidelities defined as

$$\mathcal{F}_{\text{HF}}(t) = \mathbb{P}^\mu(\mathcal{E}_{\text{HF}}(X; \alpha_{\text{HF}}) \leq t) \quad \text{and} \quad \mathcal{F}_{\text{LF}}(t) = \mathbb{P}^\mu(\mathcal{E}_{\text{LF}}(X; \alpha_{\text{LF}}) \leq t),$$

which can be estimated using the empirical distributions given by the available pilot sample $\{x_n\}_{n=1}^N$. Notice that the cumulative distribution functions satisfy

$$(\mathcal{F}_{\text{HF}} \circ \mathcal{E}_{\text{HF}}(\cdot; \alpha_{\text{HF}}))_{\#}\mu = \mathcal{U}([0, 1]) \quad \text{and} \quad (\mathcal{F}_{\text{LF}} \circ \mathcal{E}_{\text{LF}}(\cdot; \alpha_{\text{LF}}))_{\#}\mu = \mathcal{U}([0, 1]), \quad (3.7)$$

where $\#$ denotes the push-forward measure, and which means that

$$\mathcal{F}_{\text{HF}}(\mathcal{E}_{\text{HF}}(X; \alpha_{\text{HF}})) \sim \mathcal{U}([0, 1]) \quad \text{and} \quad \mathcal{F}_{\text{LF}}(\mathcal{E}_{\text{LF}}(X; \alpha_{\text{LF}})) \sim \mathcal{U}([0, 1]).$$

Therefore, we can define the modified low-fidelity model using the uniformly distributed shared space as a bridge between the two fidelities

$$\tilde{\mathcal{Q}}_{\text{LF}}(x) = \mathcal{Q}_{\text{LF}}(\mathcal{D}_{\text{LF}}(\mathcal{F}_{\text{LF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{E}_{\text{HF}}(x; \alpha_{\text{HF}}))); \beta_{\text{LF}})), \quad (3.8)$$

which can then replace the original \mathcal{Q}_{LF} in the definition of the multifidelity Monte Carlo estimator in equation (3.5). A schematic visualization of the presented pipeline is given in Figure 2. The inverse of the cumulative distribution function $\mathcal{F}_{\text{LF}}^{-1}$ stands for the generalized inverse defined as

$$\mathcal{F}_{\text{LF}}^{-1}(u) = \inf\{t \in \mathbb{R} : \mathcal{F}_{\text{LF}}(t) \geq u\}, \quad \text{for all } u \in [0, 1].$$

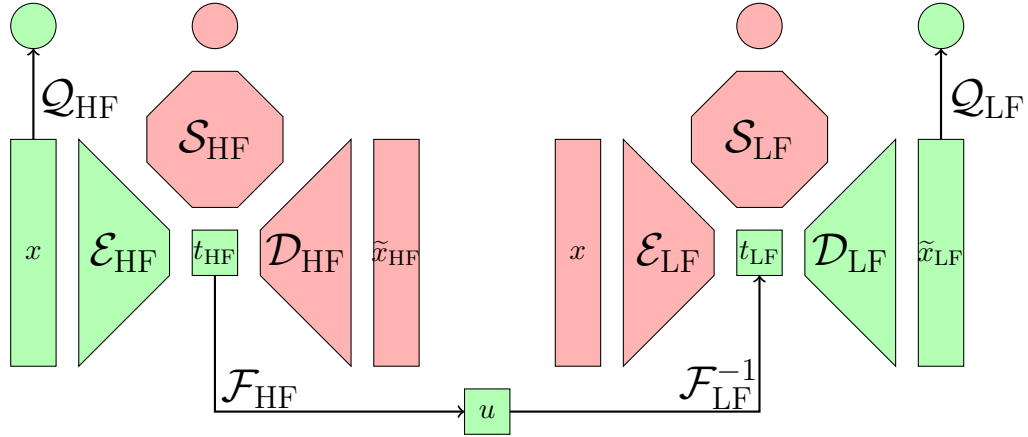


Figure 2: Schematic representation of the multifidelity uncertainty propagation pipeline based on NeurAM.

First, we notice that the estimator remains unbiased since we did not modify the high-fidelity model \mathcal{Q}_{HF} . Then, based on the ideas in [41], we expect the modified low-fidelity model to be better correlated with the high-fidelity model, and therefore we aim to achieve a smaller variance for the multifidelity Monte Carlo estimator.

Remark 3.1. In this section, for simplicity, we outlined the methodology assuming the high-fidelity and low-fidelity models to have the same parameterization. However, we note that the presented approach is also able to handle models with dissimilar parameterization due to the reduction to a shared one-dimensional manifold.

In the next section we consider the idealized setting given by the proof of Proposition 2.1, and show that, at least in this case, the new correlation

$$\tilde{\rho} = \frac{\text{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \tilde{\mathcal{Q}}_{\text{LF}}(X))}{\sqrt{\text{Var}^\mu[\mathcal{Q}_{\text{HF}}(X)] \text{Var}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)]}} \quad (3.9)$$

is indeed better than the original correlation ρ given in (3.4). This gives a theoretical guarantee under specific conditions, and therefore justifies the approach described in this section.

3.2.1 Analysis under idealized setting

In this section we assume to reach a loss function equal to zero, and we consider the global minimizer given in the proof of Proposition 2.1. In particular, let

$$\mathcal{S}_{\text{HF}} = \mathcal{I}, \quad \mathcal{E}_{\text{HF}} = \mathcal{Q}_{\text{HF}}, \quad \mathcal{Q}_{\text{HF}} \circ \mathcal{D}_{\text{HF}} = \mathcal{I}, \quad \text{and} \quad \mathcal{S}_{\text{LF}} = \mathcal{I}, \quad \mathcal{E}_{\text{LF}} = \mathcal{Q}_{\text{LF}}, \quad \mathcal{Q}_{\text{LF}} \circ \mathcal{D}_{\text{LF}} = \mathcal{I},$$

which, due to equation (3.8), gives

$$\tilde{\mathcal{Q}}_{\text{LF}}(x) = \mathcal{F}_{\text{LF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{Q}_{\text{HF}}(x))). \quad (3.10)$$

Moreover, let us assume that both \mathcal{F}_{HF} and \mathcal{F}_{LF} are invertible, and, without loss of generality, that $\rho \geq 0$. In fact, if $\rho < 0$, then we can always define $\tilde{\mathcal{Q}}_{\text{LF}}$ starting from $-\mathcal{Q}_{\text{LF}}$ instead of \mathcal{Q}_{LF} . In the next result we show that, in this case, the mean and the variance of the model are not affected, even if we are considering the reduction to its one-dimensional NeurAM.

Lemma 3.2. *Let $\tilde{\mathcal{Q}}_{\text{LF}}$ be defined in equation (3.10). Then, we have*

$$\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)] = \mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)] \quad \text{and} \quad \text{Var}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)] = \text{Var}^\mu[\mathcal{Q}_{\text{LF}}(X)].$$

Proof. First, let us rewrite the variances in terms of first and second moments

$$\begin{aligned} \text{Var}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)] &= \mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)^2] - \mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)]^2, \\ \text{Var}^\mu[\mathcal{Q}_{\text{LF}}(X)] &= \mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)^2] - \mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)]^2, \end{aligned} \quad (3.11)$$

which imply that we need to show that

$$\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)^k] = \mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)^k],$$

for $k = 1, 2$. We then have

$$\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)^k] = \int \tilde{\mathcal{Q}}_{\text{LF}}(x)^k d\mu(x) = \int \mathcal{F}_{\text{LF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{Q}_{\text{HF}}(x)))^k d\mu(x),$$

which due to (3.7) with $\mathcal{E}_{\text{HF}} = \mathcal{Q}_{\text{HF}}$ implies

$$\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)^k] = \int \mathcal{F}_{\text{LF}}^{-1}(u)^k d(\mathcal{F}_{\text{HF}} \circ \mathcal{Q}_{\text{HF}})_\# \mu(y) = \int \mathcal{F}_{\text{LF}}^{-1}(u)^k d\mathcal{U}(y) = \int q^k d\mathcal{F}_{\text{LF}\#}^{-1}\mathcal{U}(q),$$

where $\#$ stands for the push-forward measure and \mathcal{U} for the uniform distribution $\mathcal{U}([0, 1])$. Finally, using again equation (3.7) with $\mathcal{E}_{\text{LF}} = \mathcal{Q}_{\text{LF}}$, we obtain

$$\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)^k] = \int q^k d\mathcal{Q}_{\text{LF}\#} \mu(q) = \int \mathcal{Q}_{\text{LF}}(x)^k d\mu(x) = \mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)^k],$$

which, together with equations (3.11), yields the desired result. \square

We can now state the main result of this section: the modified low-fidelity model $\tilde{\mathcal{Q}}_{\text{LF}}$ is better correlated with the high-fidelity model \mathcal{Q}_{HF} than the original low-fidelity \mathcal{Q}_{LF} .

Theorem 3.3. *Let $\tilde{\mathcal{Q}}_{\text{LF}}$ be defined as in equation (3.10), and let the correlations ρ and $\tilde{\rho}$ be given in (3.4) and (3.9), respectively. Then, it holds*

$$\tilde{\rho} \geq \rho.$$

Proof. First, by Lemma 3.2, notice that $\tilde{\rho} \geq \rho$ if and only if

$$\text{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \tilde{\mathcal{Q}}_{\text{LF}}(X)) \geq \text{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \mathcal{Q}_{\text{LF}}(X)). \quad (3.12)$$

Then, rewriting the covariances as

$$\begin{aligned} \text{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \tilde{\mathcal{Q}}_{\text{LF}}(X)) &= \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\tilde{\mathcal{Q}}_{\text{LF}}(X)] - \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)]\mathbb{E}^\mu[\tilde{\mathcal{Q}}_{\text{LF}}(X)], \\ \text{Cov}^\mu(\mathcal{Q}_{\text{HF}}(X), \mathcal{Q}_{\text{LF}}(X)) &= \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\mathcal{Q}_{\text{LF}}(X)] - \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)]\mathbb{E}^\mu[\mathcal{Q}_{\text{LF}}(X)], \end{aligned}$$

and due to Lemma 3.2, we deduce that (3.12) is satisfied if and only if

$$\mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\tilde{\mathcal{Q}}_{\text{LF}}(X)] \geq \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\mathcal{Q}_{\text{LF}}(X)].$$

Let us now rewrite the left-hand side using the fact that the high-fidelity model can be rewritten as

$$\tilde{\mathcal{Q}}_{\text{HF}}(x) = \mathcal{F}_{\text{HF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{Q}_{\text{HF}}(x))).$$

In particular, by equation (3.7) with $\mathcal{E}_{\text{HF}} = \mathcal{Q}_{\text{HF}}$, we have

$$\begin{aligned} \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\tilde{\mathcal{Q}}_{\text{LF}}(X)] &= \int \mathcal{F}_{\text{HF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{Q}_{\text{HF}}(x)))\mathcal{F}_{\text{LF}}^{-1}(\mathcal{F}_{\text{HF}}(\mathcal{Q}_{\text{HF}}(x))) \, d\mu(x), \\ &= \int \mathcal{F}_{\text{HF}}^{-1}(u)\mathcal{F}_{\text{LF}}^{-1}(u) \, d(\mathcal{F}_{\text{HF}} \circ \mathcal{Q}_{\text{HF}})_\# \mu(u) \\ &= \int \mathcal{F}_{\text{HF}}^{-1}(u)\mathcal{F}_{\text{LF}}^{-1}(u) \, d\mathcal{U}(u) \\ &= \mathbb{E}^\mathcal{U}[\mathcal{F}_{\text{HF}}^{-1}(U)\mathcal{F}_{\text{LF}}^{-1}(U)], \end{aligned}$$

where the superscript \mathcal{U} denotes the fact that the expectation is computed with respect to the uniform distribution $\mathcal{U}([0, 1])$. Therefore, in order to conclude the proof, we have to show that

$$\mathbb{E}^\mathcal{U}[\mathcal{F}_{\text{HF}}^{-1}(U)\mathcal{F}_{\text{LF}}^{-1}(U)] \geq \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\mathcal{Q}_{\text{LF}}(X)]. \quad (3.13)$$

We analyze the two sides separately, and we first consider the left-hand side. The expectation can be rewritten as

$$\begin{aligned} \mathbb{E}^\mathcal{U}[\mathcal{F}_{\text{HF}}^{-1}(U)\mathcal{F}_{\text{LF}}^{-1}(U)] &= \int_0^\infty \int_0^\infty \mathbb{P}^\mathcal{U}(\mathcal{F}_{\text{HF}}^{-1}(U) \geq t, \mathcal{F}_{\text{LF}}^{-1}(U) \geq s) \, dt \, ds \\ &\quad + \int_{-\infty}^0 \int_{-\infty}^0 \mathbb{P}^\mathcal{U}(\mathcal{F}_{\text{HF}}^{-1}(U) \leq t, \mathcal{F}_{\text{LF}}^{-1}(U) \leq s) \, dt \, ds \\ &\quad - \int_{-\infty}^0 \int_0^\infty \mathbb{P}^\mathcal{U}(\mathcal{F}_{\text{HF}}^{-1}(U) \geq t, \mathcal{F}_{\text{LF}}^{-1}(U) \leq s) \, dt \, ds \\ &\quad - \int_0^\infty \int_{-\infty}^0 \mathbb{P}^\mathcal{U}(\mathcal{F}_{\text{HF}}^{-1}(U) \leq t, \mathcal{F}_{\text{LF}}^{-1}(U) \geq s) \, dt \, ds, \end{aligned}$$

which, since \mathcal{F}_{HF} and \mathcal{F}_{LF} are strictly increasing, implies

$$\begin{aligned}\mathbb{E}^{\mathcal{U}}[\mathcal{F}_{\text{HF}}^{-1}(U)\mathcal{F}_{\text{LF}}^{-1}(U)] &= \int_0^\infty \int_0^\infty \mathbb{P}^{\mathcal{U}}(U \geq \mathcal{F}_{\text{HF}}(t), U \geq \mathcal{F}_{\text{LF}}(s)) dt ds \\ &\quad + \int_{-\infty}^0 \int_{-\infty}^0 \mathbb{P}^{\mathcal{U}}(U \leq \mathcal{F}_{\text{HF}}(t), U \leq \mathcal{F}_{\text{LF}}(s)) dt ds \\ &\quad - \int_{-\infty}^0 \int_0^\infty \mathbb{P}^{\mathcal{U}}(U \geq \mathcal{F}_{\text{HF}}(t), U \leq \mathcal{F}_{\text{LF}}(s)) dt ds \\ &\quad - \int_0^\infty \int_{-\infty}^0 \mathbb{P}^{\mathcal{U}}(U \leq \mathcal{F}_{\text{HF}}(t), U \geq \mathcal{F}_{\text{LF}}(s)) dt ds.\end{aligned}$$

Then, using the fact that $U \sim \mathcal{U}([0, 1])$, we obtain

$$\begin{aligned}\mathbb{E}^{\mathcal{U}}[\mathcal{F}_{\text{HF}}^{-1}(U)\mathcal{F}_{\text{LF}}^{-1}(U)] &= \int_0^\infty \int_0^\infty (1 - \max\{\mathcal{F}_{\text{HF}}(t), \mathcal{F}_{\text{LF}}(s)\}) dt ds \\ &\quad + \int_{-\infty}^0 \int_{-\infty}^0 \min\{\mathcal{F}_{\text{HF}}(t), \mathcal{F}_{\text{LF}}(s)\} dt ds \\ &\quad - \int_{-\infty}^0 \int_0^\infty \max\{0, \mathcal{F}_{\text{LF}}(s) - \mathcal{F}_{\text{HF}}(t)\} dt ds \\ &\quad - \int_0^\infty \int_{-\infty}^0 \max\{0, \mathcal{F}_{\text{HF}}(t) - \mathcal{F}_{\text{LF}}(s)\} dt ds.\end{aligned}\tag{3.14}$$

On the other hand, the right-hand side of (3.13) reads

$$\begin{aligned}\mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)\mathcal{Q}_{\text{LF}}(X)] &= \int_0^\infty \int_0^\infty \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \geq t, \mathcal{Q}_{\text{LF}}(X) \geq s) dt ds \\ &\quad + \int_{-\infty}^0 \int_{-\infty}^0 \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \leq s) dt ds \\ &\quad - \int_{-\infty}^0 \int_0^\infty \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \geq t, \mathcal{Q}_{\text{LF}}(X) \leq s) dt ds \\ &\quad - \int_0^\infty \int_{-\infty}^0 \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \geq s) dt ds.\end{aligned}\tag{3.15}$$

We now compare each term in the right-hand side of equations (3.14) and (3.15) separately. First, we have

$$\begin{aligned}\max\{\mathcal{F}_{\text{HF}}(t), \mathcal{F}_{\text{LF}}(s)\} &= \max\{\mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t), \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s)\} \\ &\leq \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t) + \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s) \\ &\quad - \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \leq s),\end{aligned}$$

which implies

$$\max\{\mathcal{F}_{\text{HF}}(t), \mathcal{F}_{\text{LF}}(s)\} \geq \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \geq t, \mathcal{Q}_{\text{LF}}(X) \geq s).\tag{3.16}$$

Then, for the second term we find

$$\begin{aligned}\min\{\mathcal{F}_{\text{HF}}(t), \mathcal{F}_{\text{LF}}(s)\} &= \min\{\mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t), \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s)\} \\ &\geq \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \leq s).\end{aligned}\tag{3.17}$$

For the third term we have

$$\begin{aligned}\mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \geq t, \mathcal{Q}_{\text{LF}}(X) \leq s) &= \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s) - \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \leq s) \\ &\geq \max\{0, \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s) - \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t)\} \\ &= \max\{0, \mathcal{F}_{\text{LF}}(s) - \mathcal{F}_{\text{HF}}(t)\},\end{aligned}\tag{3.18}$$

and similarly for the fourth term

$$\begin{aligned}
\mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \geq s) &= \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t) - \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t, \mathcal{Q}_{\text{LF}}(X) \leq s) \\
&\geq \max\{0, \mathbb{P}^\mu(\mathcal{Q}_{\text{HF}}(X) \leq t) - \mathbb{P}^\mu(\mathcal{Q}_{\text{LF}}(X) \leq s)\} \\
&= \max\{0, \mathcal{F}_{\text{HF}}(t) - \mathcal{F}_{\text{LF}}(s)\}.
\end{aligned} \tag{3.19}$$

Finally, collecting the bounds (3.16), (3.17), (3.18), (3.19) together with the decompositions (3.14), (3.15), we deduce the inequality (3.13), which implies the desired result. \square

Remark 3.4. The analysis in this section is performed under the assumption that the neural active manifold is constructed using a global minimizer of the loss function, and hence we put ourselves in the idealized framework in which the models can be reconstructed exactly from the one-dimensional NeurAM. Even if this cannot be achieved in practice, this theoretical analysis gives a formal justification for why it is worth modifying the low-fidelity model, i.e., draw new samples from the shared space obtained from the NeurAM of the two fidelities. Therefore, we remark that, even if we cannot achieve the best correlation $\tilde{\rho}$ in practice, our methodology, in most of the cases, gives a new correlation that is better than the original correlation ρ , as we observe in the following numerical experiments. Finally, notice that the ideal correlation obtained from equation (3.13), i.e.,

$$\tilde{\rho} = \frac{\text{Cov}^\mathcal{U}(\mathcal{F}_{\text{HF}}^{-1}(U), \mathcal{F}_{\text{LF}}^{-1}(U))}{\sqrt{\text{Var}^\mathcal{U}[\mathcal{F}_{\text{HF}}^{-1}(U)] \text{Var}^\mathcal{U}[\mathcal{F}_{\text{LF}}^{-1}(U)]}}, \tag{3.20}$$

can be used as a reference value for the new correlation that we can get in practice, since it is computed in the idealized framework where the reduction to one dimension still allows us to reconstruct the original model exactly.

4 Numerical examples

In this section we apply our algorithm to construct a neural active manifold for several models. We first consider simple two-dimensional analytical functions to demonstrate the performance of NeurAM while providing intuition on the process of reducing the problem dimensionality by directly plotting the active manifold. Two computationally challenging test cases are then analyzed to demonstrate that the approach is viable for realistic problems. The first problem we consider is the so-called Hartmann problem [37], which has been used previously as a test case for other dimensionality reduction strategies, such as AS [17] and AM [7]. We also study a more complex cardiac electrophysiology model with biphasic response, requiring the solution of a preliminary classification task. Finally, we compare the multifidelity UQ pipeline introduced in the previous section with standard multifidelity Monte Carlo estimators for model with multiple inputs, and perform sensitivity analysis. We remark that, for each numerical example, we use a different set of hyperparameters (layers and neurons) for the neural networks corresponding to the encoder \mathcal{E} , decoder \mathcal{D} , and surrogate model \mathcal{S} . All the hyperparameters are optimized using Optuna [1] monitoring the validation loss (20% of the dataset) for 100 iterations. In particular, we constrain the number of layers to be in $\{1, \dots, 4\}$ and the number of neurons per layer to be in $\{1, \dots, 16\}$, and we train all the networks for 10000 epochs. Hence, we emphasize that the resulting neural networks have a limited size, implying a minimal computational cost for their training.

4.1 Two-dimensional models

We analyze several two-dimensional synthetic test cases to assess the performance of the proposed approach and visualize the neural active manifold.

4.1.1 Parabolic neural active manifold

We first consider the model $Q: \mathbb{R}^2 \rightarrow \mathbb{R}$ defined as

$$Q(x) = x_1^2 + x_2,$$

and let the distribution of the input parameters be $\mu = \mathcal{U}([0, 1]^2)$. Notice that this is an example where the global minimizer given by the proof of Proposition 2.1 can be computed analytically. In particular, we have

$$\mathcal{S}^*(t) = t, \quad \mathcal{E}^*(x) = x_1^2 + x_2, \quad \mathcal{D}^*(t) = \left[\sqrt{\frac{t}{2}} \quad \frac{t}{2} \right]^\top, \quad (4.1)$$

which implies that $\mathcal{L}(\mathcal{E}^*, \mathcal{D}^*, \mathcal{S}^*) = 0$. The NeurAM is then parameterized by the decoder $\mathcal{D}^*(t)$ for all values of t in the interval

$$\mathcal{T} = \left[\min_{x_1, x_2 \in [0, 1]} (x_1^2 + x_2), \max_{x_1, x_2 \in [0, 1]} (x_1^2 + x_2) \right] = [0, 2],$$

and is represented by a parabola of the form $x_2 = x_1^2$ for all $x_1 \in [0, 1]$. In Figure 3 we illustrate the NeurAM that we computed analytically on top of the contour plot of the model, and we show through red arrows how the points are projected on the manifold by the autoencoder, i.e., computing $\mathcal{D}(\mathcal{E}(x))$. Moreover, from the parity plot on the right, we observe that the exact and surrogate models coincide, i.e., $Q(x) = Q_S(x) = \mathcal{S}(\mathcal{E}(x))$, resulting in the entire model variance being captured.

Let us now slightly modify the function Q , and consider a more complex model $\mathcal{Q}: \mathbb{R}^2 \rightarrow \mathbb{R}$ given by

$$\mathcal{Q}(x) = \sin(Q(x)) = \sin(x_1^2 + x_2),$$

for which the NeurAM remains the same if we choose as surrogate model $\mathcal{S}(t) = \sin(t)$. We then run our algorithm with an increasing number of training samples $N = 10, 30, 50, 100, 500, 1000$, and compute the mean absolute error (MAE) and the mean squared error (MSE) for

$$e_1(x) = \mathcal{Q}(x) - \mathcal{Q}(\mathcal{D}(\mathcal{E}(x))), \quad e_2(x) = \mathcal{Q}(x) - \mathcal{Q}_S(x), \quad (4.2)$$

over 100 repetitions. Note that e_1 only measures the error due to the dimensionality reduction, while e_2 quantifies the error in the surrogate model. The MAE and MSE for e_1 and e_2 are computed using a testing set with 1000 randomly selected samples. In Figure 4 we present the resulting error trends, using error bars to represent standard deviations produced by repeated runs. As expected, both errors decrease with the increase of the training dataset, until they reach a plateau. We remark that the standard deviation of the model \mathcal{Q} over its domain is

$$\sqrt{\mathbb{E}^\mu \left[(\mathcal{Q}(X) - \mathbb{E}^\mu[\mathcal{Q}(X)])^2 \right]} \simeq 0.258,$$

implying that the relative errors are approximately of the same order of the absolute errors reported in Figure 4.

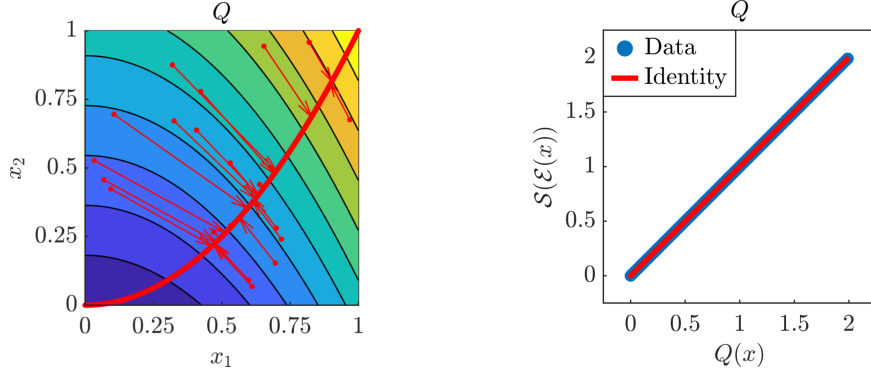


Figure 3: Analytical results for the two-dimensional model Q . Left: projection of input samples on the one-dimensional NeurAM. Right: parity plot between the exact and surrogate models.

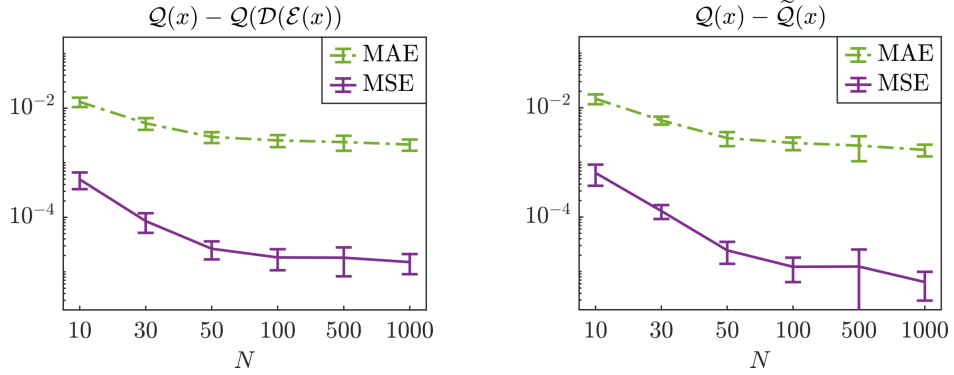


Figure 4: Approximation errors (MAE and MSE) in equation (4.2) as functions of the size of the training set N for the two-dimensional model Q . Left: error e_1 due to the dimensionality reduction. Right: error e_2 of the surrogate model.

4.1.2 Comparison with active manifolds and active subspaces

We consider the following two-dimensional models that have been used to test AM in [7]

$$\begin{aligned} Q_1(x) &= e^{x_2 - x_1^2}, \\ Q_2(x) &= x_1^2 + x_2^2, \\ Q_3(x) &= x_1^3 + x_2^3 + 0.2x_1 + 0.6x_2, \end{aligned}$$

and we let the distribution of the input parameters be $\mu = \mathcal{U}([-1, 1]^2)$. We first compute NeurAM using $N = 1000$ samples. The results in Figure 5 show the projections determined through dimensionality reduction (red arrows), while the parity plots show how the surrogate model on the latent space, i.e., as a function of the reduced variable, is able to capture practically the entire variability for all the models. We remark that these plots are obtained for one possible realization of the NeurAM, meaning that the method can generate alternative NeurAM with similar overall results.

We then compare the performance of NeurAM with results for AS and AM, as reported in [7, Table 1]. We run our algorithm 100 times using $N = 1000$ training samples, and we compute the errors using equation (4.2) on a testing set of size 1000. Even though our approach presents a slightly larger variability, which is mainly due to the fact that the solution of the optimization algorithm is not unique, Figure 6 shows that NeurAM

is consistently able to achieve smaller errors than competing algorithms. Moreover, our method is always able to find a lower-dimensional representation, unlike AM for which a percentage of points in the domain is left out as shown in the table in Figure 6, meaning that we can always compute the projection of a point on the lower-dimensional manifold, similarly to AS. Finally, NeurAM requires fewer training points (1000 for NeurAM and 8000 for AM and AS as stated in [7]), and, more crucially, does not need the computation of the gradient, which is instead necessary for both AS and AM.

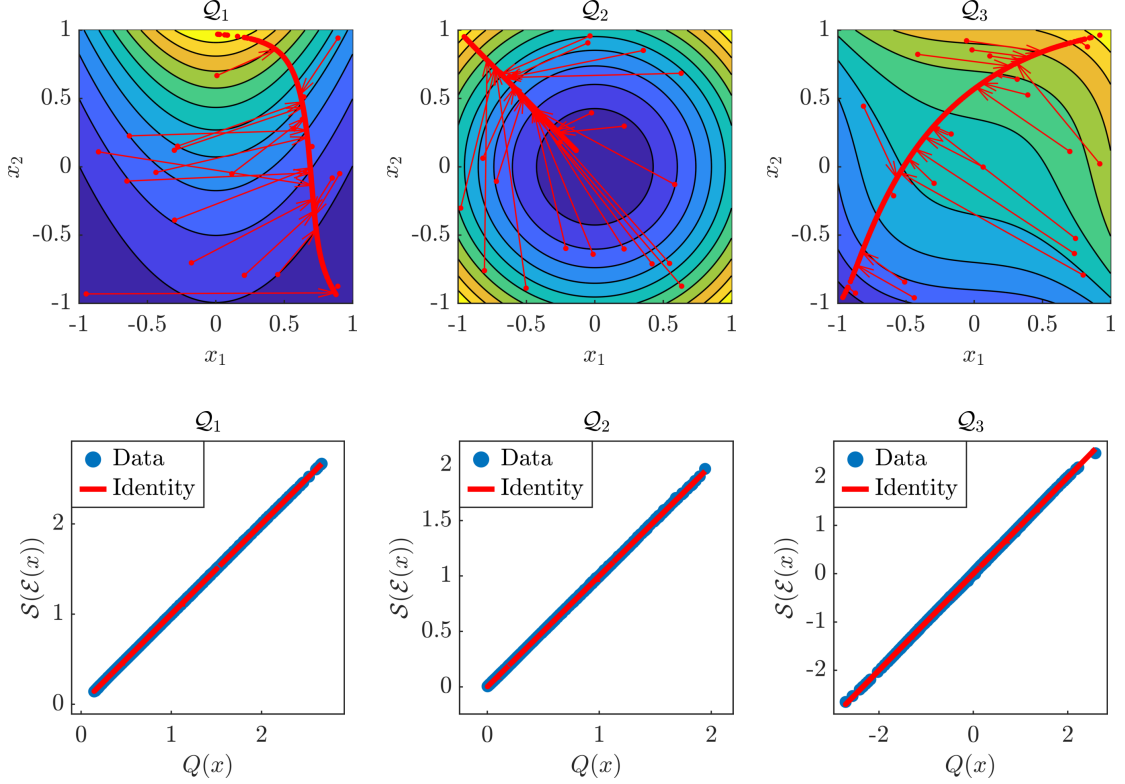


Figure 5: Results for the two-dimensional models $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$, for one realization of the NeurAM. Top: projection of input samples on the one-dimensional NeurAM. Bottom: parity plot between the exact and surrogate models.

4.1.3 NeurAM for multifidelity uncertainty propagation

In this section we show another advantage of NeurAM, the ability to enhance the performance of multifidelity Monte Carlo estimators, without any additional modification or computational cost. Specifically, we consider the following high-fidelity and low-fidelity models

$$\begin{aligned}\mathcal{Q}_{\text{HF}}(x) &= e^{0.7x_1+0.3x_2} + 0.15 \sin(2\pi x_1), \\ \mathcal{Q}_{\text{LF}}(x) &= e^{0.01x_1+0.99x_2} + 0.15 \sin(3\pi x_2),\end{aligned}$$

proposed in [16] to test the application of AS to multifidelity UQ, and then further employed in [41] in the context of nonlinear dimensionality reduction. We assume $x \sim \mu = \mathcal{U}([-1, 1]^2)$, and focus on the estimation of the quantity of interest q expressed as

$$q = \mathbb{E}^\mu[\mathcal{Q}_{\text{HF}}(X)] = \frac{25}{21} \left(e^{-1} - e^{-\frac{2}{5}} - e^{\frac{2}{5}} + e \right).$$

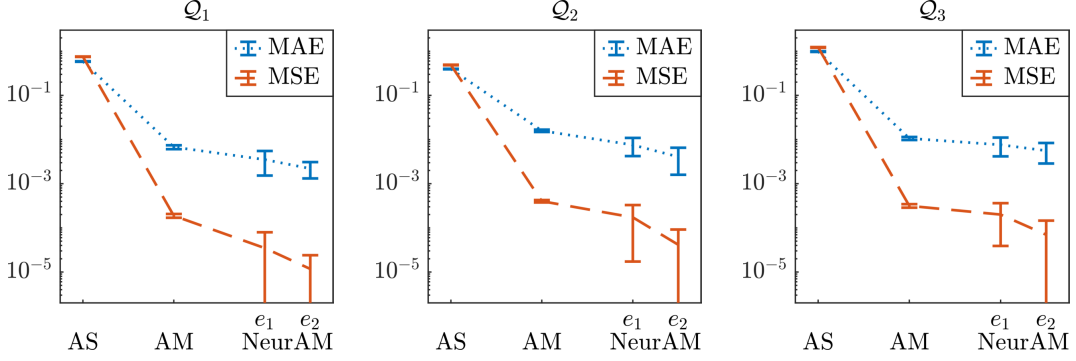


Figure 6: Comparison of approximation errors (MAE and MSE) for AS, AM, and NeurAM, applied to the two-dimensional models $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$. Top: plot of the approximation errors, where the values for AS and AM are reported from [7, Table 1], and the errors e_1 and e_2 for NeurAM are defined in equation (4.2). These results consider 8000 training samples for AS and AM, and 1000 samples for NeurAM. Bottom: percentage of test points for which the algorithm successfully found an approximation.

We further assume a computational budget $\mathcal{B} = 1000$ and a cost of our low-fidelity model equal to $\mathcal{C}_{\text{LF}} = 0.01\mathcal{C}_{\text{HF}}$ ($w = 0.01$) to mirror cost differences in realistic applications.

We compute NeurAM for both the high-fidelity and low-fidelity models separately using $N = 1000$ training samples, and present the results in Figure 7, which shows that the variability for both models is entirely captured. We remark that, even if we might expect a vertical NeurAM, as $\mathcal{Q}_{\text{LF}}(x)$ only varies along the vertical direction, any curve going from the bottom (minimum value) to the top (maximum value) of the domain is a valid NeurAM. What is crucial is how points outside the manifold are projected on it, which occurs horizontally, i.e., along the direction where the model response is constant. This indeed suggests the correctness of the results for this case.

We then move to the problem of estimating the quantity of interest q using a multifidelity Monte Carlo estimator. We compare multifidelity Monte Carlo from (3.5) with our modified estimator, where $\tilde{\mathcal{Q}}_{\text{LF}}$ is used instead of \mathcal{Q}_{LF} , and computed applying NeurAM to map high-fidelity to low-fidelity model inputs. We also include a single fidelity Monte Carlo estimator in our comparison as a reference. Figure 8 shows the results obtained from 100 repeated evaluations of the three estimators. We observe that the original models \mathcal{Q}_{HF} and \mathcal{Q}_{LF} are badly correlated, leading to a poor performance of the multifidelity Monte Carlo estimator, which indeed does not improve over single-fidelity Monte Carlo. Conversely, NeurAM significantly increases the correlation between the models, leading to a multifidelity estimator with reduced variance. Finally, the box plot in Figure 8 also includes the ideal correlation from equation (3.20), which seems to act as an upper bound for the correlation achievable through NeurAM.

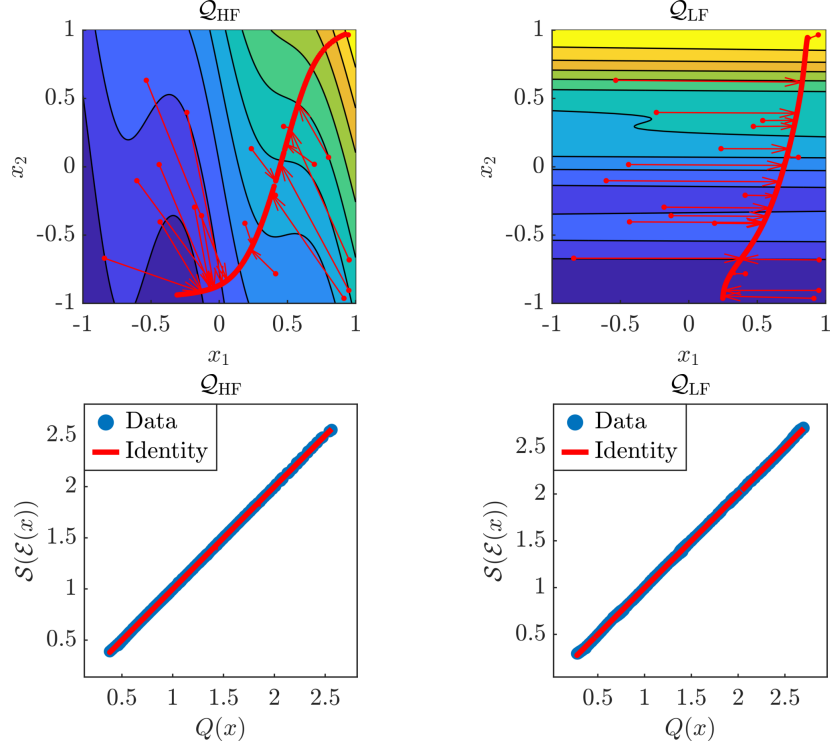


Figure 7: Results for the two-dimensional models Q_{HF} , Q_{LF} , for one realization of the NeurAM. Top: projection of input samples on the one-dimensional NeurAM. Bottom: parity plot between the exact and surrogate models.

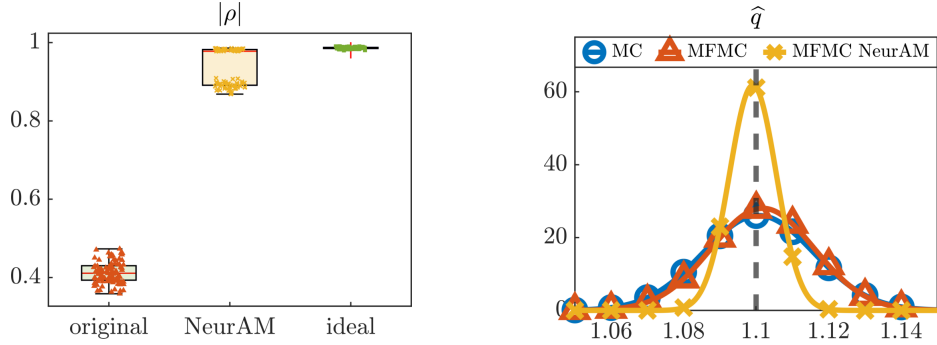


Figure 8: Numerical results for the multifidelity uncertainty propagation. Left: Pearson correlation coefficients. Right: approximated distributions from single-fidelity and multifidelity Monte Carlo estimators. The dashed line indicates the exact value of the quantity of interest q .

4.1.4 Sensitivity analysis for the two-dimensional models

Consider the model Q with the parabolic NeurAM in Section 4.1.1, for which we can compute the sensitivity indices analytically. Substituting the expressions in equation (4.1) into the local indices (3.1) we obtain

$$\theta_1(t) = \frac{2t}{2t+1} \quad \text{and} \quad \theta_2(t) = \frac{1}{2t+1},$$

for $t \in [0, 2]$ which implies that the first input is more important than the second if $t > 1/2$, while the opposite is true for $t \leq 1/2$. Moreover, computing the integrals in equation (3.3)

leads to the global indices

$$\Theta_1 = \frac{\int_0^2 \frac{2t}{2t+1} \sqrt{\frac{2t+1}{8t}} dt}{\int_0^2 \sqrt{\frac{2t+1}{8t}} dt} = \frac{\int_0^2 \sqrt{\frac{4t}{2t+1}} dt}{\int_0^2 \sqrt{\frac{2t+1}{t}} dt} = \frac{4\sqrt{5} - 2\log(2 + \sqrt{5})}{4\sqrt{5} + 2\log(2 + \sqrt{5})} \simeq 0.512,$$

$$\Theta_2 = \frac{\int_0^2 \frac{1}{2t+1} \sqrt{\frac{2t+1}{8t}} dt}{\int_0^2 \sqrt{\frac{2t+1}{8t}} dt} = \frac{\int_0^2 \sqrt{\frac{1}{t(2t+1)}} dt}{\int_0^2 \sqrt{\frac{2t+1}{t}} dt} = \frac{4\log(2 + \sqrt{5})}{4\sqrt{5} + 2\log(2 + \sqrt{5})} \simeq 0.488.$$

We then move to the models $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$, and $\mathcal{Q}_{\text{HF}}, \mathcal{Q}_{\text{LF}}$ in Sections 4.1.2 and 4.1.3, respectively. Both local and global indices are reported in Figure 9, where we also compare the results with the first order Sobol' sensitivity indices, even if we do not expect perfect agreement since our sensitivity indices are based on derivatives, while Sobol' indices are based on an analysis of variance. For the last \mathcal{Q}_{LF} model, where the only significant variable is x_2 as shown in Figure 7, the indices are constantly equal to 0 and 1 for x_1 and x_2 , respectively, as expected. For the second model \mathcal{Q}_2 , where one would expect equal importance for both the variables, we notice that our local and global indices slightly diverge from the value 0.5. This is due to the fact that the indices are dependent on the single realization of the NeurAM. Nevertheless, for all the realizations, the values oscillate around 0.5, implying that both the variables are important. Moreover, computing the average global indices over 100 runs, we obtain $\bar{\Theta}_1 = 0.499$ and $\bar{\Theta}_2 = 0.501$, as expected. For the other models, the importance of the input variables as computed from NeurAM and Sobol' indices is consistent, with the only exception of \mathcal{Q}_1 for which the interaction between the two components is more relevant. In particular, we observe a better agreement between the two types of indices for $\mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_{\text{LF}}$, where the interaction between the variables is limited. Our indices are therefore more suitable to correctly rank the variables in the presence of a negligible interaction. Extensions of NeurAM to quantify sensitivity accounting for interaction among inputs are left to future work.

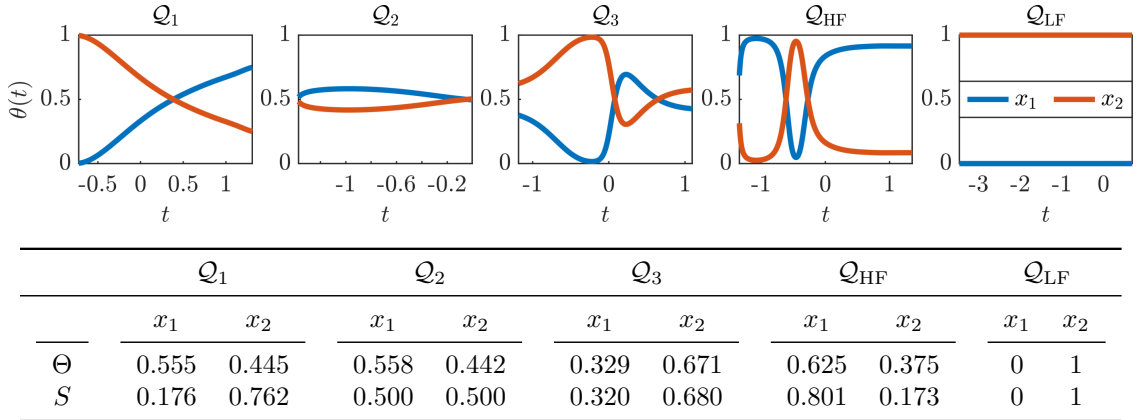


Figure 9: Sensitivity analysis for one realization of the NeurAM for the two-dimensional models $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_{\text{HF}}, \mathcal{Q}_{\text{LF}}$. Top: local indices $\theta(t)$ for both x_1 (blue) and x_2 (red). Bottom: global indices Θ compared with Sobol' indices S .

4.2 Hartmann problem

In this section we consider a more complex example, the Hartmann problem, which is a verification test case for magnetohydrodynamics applications. The Hartmann problem

models a laminar flow of a conducting fluid between two parallel plates, while a magnetic field is applied in the transverse direction [37]. This problem admits an analytical solution and has been used previously to test AS and AM in [7, 17]. In particular, the two quantities of interest, i.e., the average flow velocity u and the induced magnetic field B , are given by the following expression [7, 17]

$$\begin{cases} u(\mu, \rho, \partial p_0/\partial x, \eta, B_0) = -\frac{\partial p_0}{\partial x} \frac{\eta}{B_0^2} \left[1 - \frac{B_0 l}{\sqrt{\eta \mu}} \coth \left(\frac{B_0 l}{\sqrt{\eta \mu}} \right) \right], \\ B(\mu, \rho, \partial p_0/\partial x, \eta, B_0) = \frac{\partial p_0}{\partial x} \frac{l \mu_0}{2 B_0} \left[1 - 2 \frac{\sqrt{\eta \mu}}{B_0 l} \tanh \left(\frac{B_0 l}{2 \sqrt{\eta \mu}} \right) \right], \end{cases}$$

where the magnetic constant $\mu_0 = 1$ and the length $l = 1$ are fixed, while the remaining parameters are log-uniformly distributed in the ranges given in Table 1. Note that the fluid density ρ does not actually appear in the equations for u and B . We first normalize the input parameters in $[-1, 1]^5$, and run our algorithm using $N = 1000$ training samples. Since we cannot visualize the NeurAM for a five-dimensional problem, Figure 10 only shows a parity plot for both the average flow velocity u and the induced magnetic field B . We observe that the model variability is almost completely captured by NeurAM, but, unlike the two-dimensional models in the previous sections, a few points are observed to deviate from the identity.

We then compare the performance of our methodology with the numerical results for AS and AM reported in [7, Table 3]. We run NeurAM 100 times, and then compute the MAE and MSE of the errors e_1 and e_2 in (4.2) at each iteration for both u and B . The comparative plot in Figure 11 shows an improved performance of NeurAM for both QoIs, particularly the induced magnetic field B . This can be justified in light of the more complex dependence between B and the five inputs, whereas linear dimensionality reduction seems sufficient for u (see, e.g., the sensitivity indices in Figure 13).

We also use the optimally trained NeurAM for multifidelity propagation and sensitivity analysis. In this context, we assume the induced magnetic field B and the average flow velocity u to represent the low-fidelity and high-fidelity model response, respectively, and focus on computing the expectation $\bar{u} = \mathbb{E}^\nu[u]$, where ν denotes the joint distribution of the input parameters given in Table 1. In order to mimic cost differences in realistic applications, we let $w = 0.01$ be the cost ratio $\mathcal{C}_{\text{LF}}/\mathcal{C}_{\text{HF}}$ between the two fidelities. We then compute 100 estimates of \bar{u} with a computational budget $\mathcal{B} = 1000$, using single-fidelity Monte Carlo and multifidelity Monte Carlo with the original low-fidelity model or using NeurAM to determine new sample locations according to (3.8). In Figure 12 we show that the new low-fidelity sampling locations computed by NeurAM improve the correlation between B and u , and in several cases reach the ideal correlation given in Remark 3.4. As a consequence, the resulting approximated distribution of the multifidelity Monte Carlo estimator with NeurAM has a significantly smaller variance than standard multifidelity Monte Carlo.

Finally, in Figure 13 we show that NeurAM can be used for sensitivity analysis. In particular, we plot the quantities θ_i , $i = 1, \dots, 5$, given in equation (3.1) for each input. The input ranking varies along the manifold for B , while it remains approximately constant for u . For example, the dependence of B on B_0 increases along the manifold, implying a smaller contribution of $\partial p_0/\partial x$, which is the most important input parameter. On the other hand, we notice that the second most relevant parameter for u is μ , while the remaining inputs give a negligible contribution. Moreover, in both cases we observe that the index

$\theta(t)$ for ρ is constantly zero as expected, since it does not appear in the equations for u and B . In Figure 13 we finally compare the global indices Θ_i with the first order Sobol' sensitivity indices S_i computed evaluating the exact models over 2^{10} samples, showing agreement between the two rankings.

Parameter	Notation	Range	Unit
Fluid viscosity	μ	$[0.05, 0.2]$	$\text{kg m}^{-1} \text{s}^{-1}$
Fluid density	ρ	$[1, 5]$	kg m^{-3}
Applied pressure gradient	$\frac{\partial p_0}{\partial x}$	$[0.5, 3]$	$\text{kg m}^{-2} \text{s}^{-2}$
Resistivity	η	$[0.5, 3]$	$\text{kg m}^3 \text{s}^{-3} \text{A}^{-2}$
Applied magnetic field	B_0	$[0.1, 1]$	$\text{kg s}^{-2} \text{A}^{-1}$

Table 1: Parameter space for the Hartmann problem.

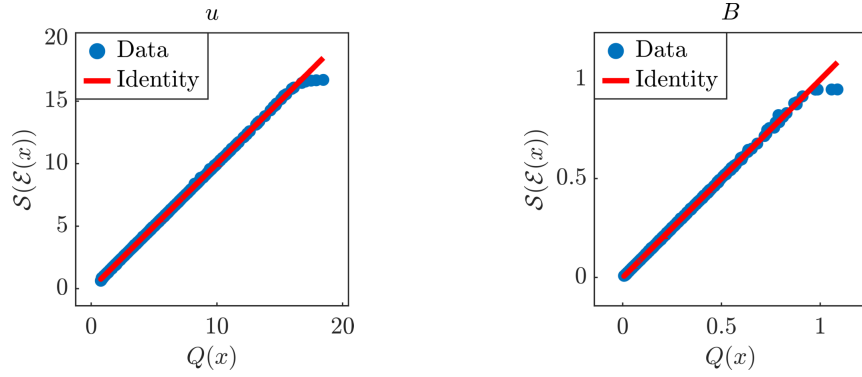


Figure 10: Parity plot between the exact and surrogate models, for one realization of the NeurAM for the Hartmann problem. Left: average flow velocity u . Right: induced magnetic field B .

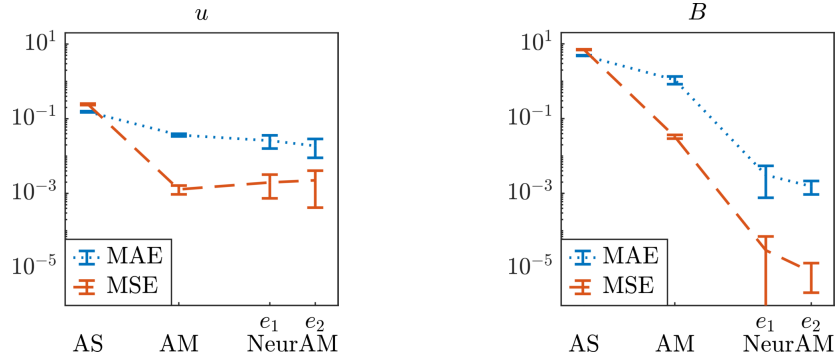


Figure 11: Comparison of approximation errors (MAE and MSE) between AS, AM and NeurAM for the Hartmann problem. The values for AS and AM are reported from [7, Table 3], and the errors e_1 and e_2 for NeurAM are defined in equation (4.2). Left: average flow velocity u . Right: induced magnetic field B .

4.3 Cardiac electrophysiology

In this section we consider a more complex model which cannot be expressed as an analytic function of the input parameters. In particular, we model cardiac electrophysiology in a

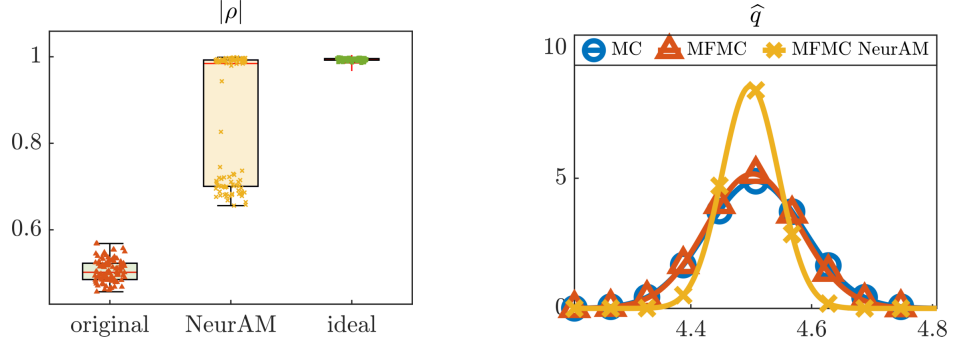
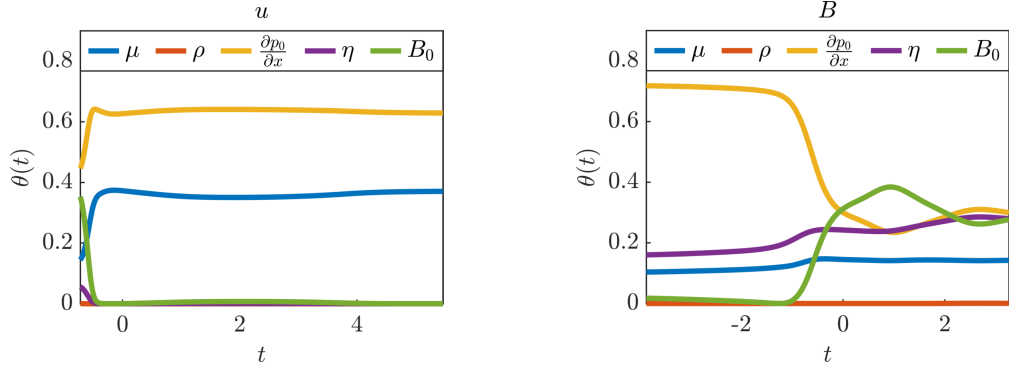


Figure 12: Multifidelity uncertainty propagation for the Hartmann problem. Left: Pearson correlation coefficients. Right: approximated distributions for the single fidelity and multifidelity Monte Carlo estimators.



	u						B				
	μ	ρ	$\partial p_0/\partial x$	η	B_0		μ	ρ	$\partial p_0/\partial x$	η	B_0
Θ	0.332	2e-5	0.619	0.007	0.043	Θ	0.140	4e-5	0.372	0.238	0.250
S	0.299	0	0.572	0.001	0.025	S	0.116	0	0.299	0.199	0.212

Figure 13: Sensitivity analysis for one realization of the NeurAM for the Hartmann problem. Left: average flow velocity u . Right: induced magnetic field B . Top: local indices $\theta(t)$ from NeurAM. Bottom: global NeurAM (Θ) and Sobol' (S) sensitivity indices.

two-dimensional square $\Omega = [0, L]^2$ of length $L = 250$ mm representing a slab of cardiac tissue using the biophysically detailed monodomain equation [9, 33], coupled with the ten Tusscher-Panfilov ionic model [39]. We formulate the model as follows

$$\left\{ \begin{array}{ll} \frac{\partial u}{\partial t} + \mathcal{I}_{\text{ion}}(u, y) - \nabla \cdot (D_{\text{iso}} \nabla u) = \mathcal{I}_{\text{app}}(\omega, \tau; \tau_{\text{box}}^{\text{stim}}) & \text{in } \Omega \times (0, T], \\ (D_{\text{M}} \nabla u) \cdot n = 0 & \text{on } \partial\Omega \times (0, T], \\ \frac{dy}{dt} = F(u, y; G_{\text{Na}}, G_{\text{CaL}}, G_{\text{Kr}}, G_{\text{Ks}}) & \text{in } \Omega \times (0, T], \\ u(\omega, 0) = u_0(\omega), y(\omega, 0) = y_0(\omega) & \text{in } \Omega. \end{array} \right. \quad (4.3)$$

The transmembrane potential u describes the propagation of the electric signal over the two-dimensional slab of cardiac tissue, the vector $y = (y_1, \dots, y_{M+P})$ defines the probability density functions of $M = 12$ gating variables, which represent the fraction of open channels

across the membrane of a single cardiomyocyte, and the concentrations of $P = 6$ relevant ionic species. Among them, sodium Na^+ , intracellular calcium Ca^{2+} and potassium K^+ play an important role in the physiological processes [3] dictating heart rhythmicity and sarcomere contractility [6]. The right hand side $F(u, y)$ defines the dynamics of the gating and concentration variables. The ionic current $\mathcal{I}_{\text{ion}}(u, y)$ models the impact of the system of ordinary differential equations at the cellular, microscopic scale on the tissue, macroscopic scale coming for the first partial differential equation. The analytical expressions of both $F(u, y)$ and $\mathcal{I}_{\text{ion}}(u, y)$ derive from the mathematical formulation of the ten Tusscher-Panfilov ionic model [39]. The right-hand side F depends also on the maximal Ca^{2+} and Na^+ current conductances G_{CaL} and G_{Na} , and the maximal rapid and slow delayed rectifier current conductances G_{Kr} and G_{Ks} . The diffusion tensor $D_{\text{iso}}I$ defines an isotropic propagation of the electric signal driven by the homogeneous isotropic conductivity D_{iso} . We impose the condition of an electrically isolated domain by prescribing homogeneous Neumann boundary conditions $\partial\Omega$, where n is the outward unit normal vector to the boundary. The action potential is triggered by a uniform current $\mathcal{I}_{\text{app}}(\omega, \tau; \tau_{\text{box}}^{\text{stim}})$ that is applied at time $\tau = 0$ s on $\{0\} \times [0, 125]$, which is the left side of the square domain, followed by another stimulus at a variable time $\tau = \tau_{\text{box}}^{\text{stim}}$ on the $[0, 125]^2$. Finally, the maximal rapid and slow delayed rectifier current conductances are fixed at $G_{\text{Kr}} = 0.153$ nS pF^{-1} and $G_{\text{Ks}} = 0.392$ nS pF^{-1} , respectively, and the final simulation time is $T = 2$ s.

We perform space discretization of the model (4.3) using \mathbb{P}_1 finite elements. The tetrahedral meshes are comprised of 20353 cells and 10400 degrees of freedom, and the average mesh size is $h = 3.5$ mm. Regarding time discretization, we first update the variables of the ionic model and then the transmembrane potential by employing an implicit-explicit numerical scheme [15, 31, 34]. Specifically, in the monodomain equation, the diffusion term is treated implicitly and the ionic term is treated explicitly. Moreover, the ionic current is discretized by means of the ionic current interpolation scheme [22]. We employ a time step size $\Delta t = 0.1$ ms with the forward Euler scheme.

In Table 2 we report descriptions, ranges and units for the four model parameters. We initially generate a dataset of 10000 electrophysiology simulations (5000 for training and 5000 for testing) by exploring the space of parameters with stratified sampling (latin hypercube sampling [27]). The quantity of interest $\mathcal{Q}(X)$ that is monitored by our method is the space integral of the transmembrane potential u at the final time T , i.e.,

$$\mathcal{Q}(X) = \int_{\Omega} u(\omega, T; X) \, d\omega,$$

where

$$X = \begin{bmatrix} G_{\text{CaL}} & G_{\text{Na}} & D_{\text{iso}} & \tau_{\text{box}}^{\text{stim}} \end{bmatrix}^{\top} \in \mathbb{R}^4$$

is the vector of inputs. We remark that all the input parameters and the quantity of interest have been normalized in the interval $[-1, 1]$. The parameter space covered by the model (4.3) enables numerical simulations exhibiting both sinus rhythm and sustained arrhythmia, as shown in Figure 14. Indeed, this test case presents a *bifurcation*, meaning that the response varies abruptly even for locations that are relatively close in the space of parameters (alternative ways to describe this behavior would be to say that the model response is *biphasic* or characterized by sharp gradients), which leads to a challenging mathematical problem for the presented dimensionality reduction approach. The bifurcation is visible in Figure 15, where we plot the histogram of the quantity of interest, and notice that \mathcal{Q} takes either values close to -1 , with a very few exceptions in the interval $(-0.9, 0)$ that are not visible in the histogram, or values approximately in the interval $(0.4, 1)$. We

therefore first perform a classification to divide the input space in two classes, based on whether the normalized quantity of interest is positive or negative, and then compute the NeurAM in each class separately. We build a neural network classifier using the training data, and then plot the results on the test set in Figure 15. We observe that the classifier is able to correctly discriminate between sinus rhythm and sustained arrhythmia in 97.68% of the cases.

We now apply our methodology to learn the NeurAM for the two classes. In Figure 16, where we plot the exact and surrogate models as functions of the reduced variable, we notice that the NeurAM is able to capture most of the variability of the model. We remark that, for the case of the sinus rhythm, the model takes only values in the neighborhood of -1 , and therefore the surrogate model is almost constant. Note that the plot on the left of Figure 16 might be misleading, as almost all the points are concentrated around $(-1, 1)$, and only a few points are left out from the identity line and give a nonnegligible error. Regarding the case of sustained arrhythmia, where there is more variability, the surrogate model is able to provide a reliable approximation, but it is not as accurate as in the test cases in Section 4.1 due to the increased complexity of the problem. In Table 3 we report the MAE and the MSE for the error e_2 in equation (4.2) given by the surrogate model. We remark that in this example we cannot compute the error e_1 given by the dimensionality reduction because we would need to run additional simulations in the projected points, which are not in the original dataset. We notice that the errors in Table 3, compared with the standard deviation σ of the model over the input domain, are particularly small for the single classes, meaning that NeurAM gives accurate surrogate models for each class. We also notice that the overall error, obtained first applying the classifier and then the surrogate model of the predicted class, is slightly larger. This is due to the fact that a wrong classification yields a large error for the particular sample, implying a larger overall error contribution. Nevertheless, the derived surrogate model still provides a reliable approximation of the real model since the majority of the data is correctly classified.

We finally perform a sensitivity analysis for the cardiac electrophysiology model, considering both classes of sinus rhythm and sustained arrhythmia. The local sensitivity indices $\theta_i(t)$ and the global sensitivity indices Θ_i are illustrated in Figure 17 for all the parameters i . We first notice that all the variables appear to be unimportant in the case of sinus rhythm, except for $\tau_{\text{box}}^{\text{stim}}$, which is the only one that affects the quantity of interest for the parameter space explored in Table 2, since it is crucial to distinguish this physiological state from the onset of possible rhythm disorders [36]. Regarding the case of sustained arrhythmia, we observe how the relevance of each input parameter varies along the NeurAM. In particular, we notice that, as reflected by the global indices in the table of Figure 17, G_{CaL} is the most important input parameter, since it plays an important role in arrhythmogenesis [3], followed by $\tau_{\text{box}}^{\text{stim}}$, D_{iso} and G_{Na} , respectively. Moreover, from the dynamic indices, we deduce that $\tau_{\text{box}}^{\text{stim}}$ is more relevant in the first and last part of the NeurAM, while D_{iso} is more important in the middle part.

5 Conclusion

In this work we introduced a novel technique for dimensionality reduction of computationally expensive models. In particular, we employed autoencoders with one-dimensional latent space to obtain a manifold in the parameter space (neural active manifold or NeurAM)

Parameter	Notation	Range	Unit
Stimulation time	$\tau_{\text{box}}^{\text{stim}}$	[300, 400]	ms
Isotropic conductivity	D_{iso}	[0.2114, 0.4757]	$\text{mm}^2 \text{ms}^{-1}$
Maximal Ca^{2+} conductance	G_{Na}	[9.892, 22.2570]	nS pF^{-1}
Maximal Na^+ conductance	G_{CaL}	[2.653e-05, 5.970e-05]	$\text{cm}^3 \text{ms}^{-1} \mu\text{F}^{-1}$

Table 2: Input parameter space for the cardiac electrophysiology model.

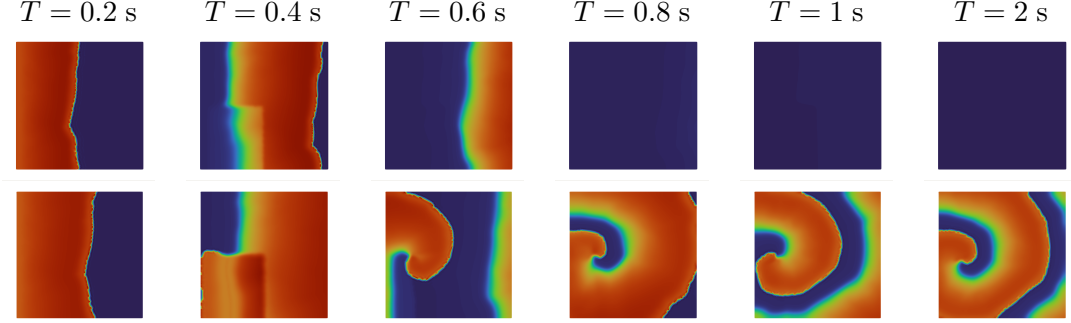


Figure 14: Examples of evolution in time of the transmembrane potential u in the cardiac electrophysiology model, for different time instants in the numerical simulation. Top: sinus rhythm. Bottom: sustained arrhythmia.

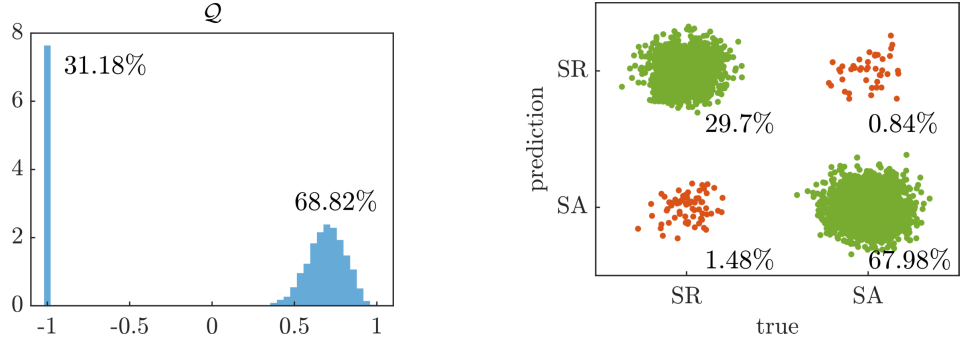


Figure 15: Classification of cardiac electrophysiology response. Left: histogram of the quantity of interest Q . Right: results from the classification algorithm.

Sinus rhythm			Sustained arrhythmia			Overall		
MAE	MSE	σ	MAE	MSE	σ	MAE	MSE	σ
0.0002	0.0039	0.0240	0.0018	0.0293	0.1127	0.0689	0.0602	0.7905

Table 3: Approximation error (MAE and MSE) of the surrogate model, compared with the standard deviation σ of the original model, for both sinus rhythm and sustained arrhythmia in the cardiac electrophysiology model. In the third column the overall error is computed employing the classifier as a first step.

capturing as much as possible of the model output variability. The autoencoder is combined with a surrogate model with inputs in the latent space that is trained at the same time as the encoder and decoder. NeurAM provides a concise one-dimensional analogue of the original computationally expensive model, and can be used to efficiently perform

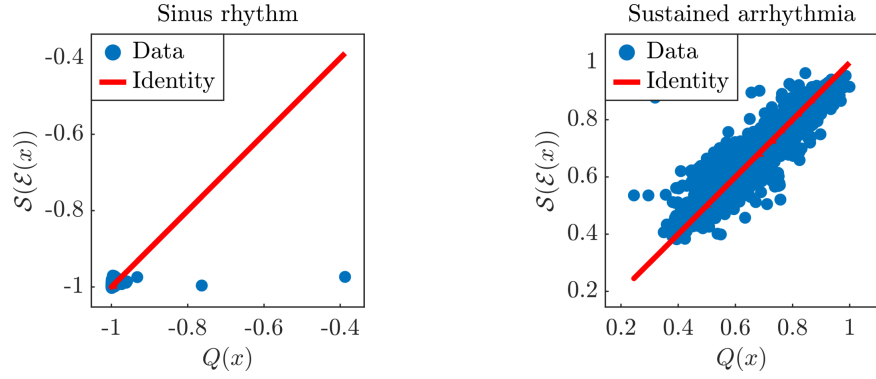


Figure 16: Parity plot between the exact and surrogate models, for one realization of the NeurAM for the cardiac electrophysiology model. Left: sinus rhythm. Right: sustained arrhythmia.

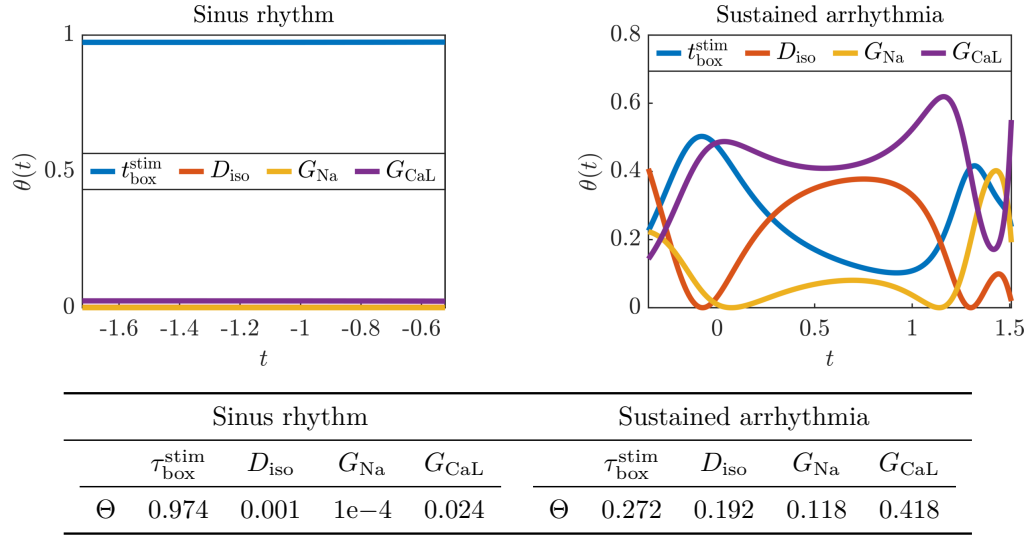


Figure 17: NeurAM sensitivity analysis for the cardiac electrophysiology model. Left: sinus rhythm. Right: sustained arrhythmia. Top: local indices $\theta(t)$. Bottom: global indices Θ .

uncertainty quantification tasks. We first showed how NeurAM can be used to perform both local and global sensitivity analysis. In particular, we proposed new local indices providing a dynamic ranking of the input parameters along the manifold. Moreover, by integrating these local indices, we derived global indices that quantify the overall ranking of the input parameters. We then focused on the problem of multifidelity uncertainty propagation, selecting a low- and high-fidelity model pair. Following [41], we used NeurAM to determine a shared space bridging the inputs of the two models, and to generate new low-fidelity input samples resulting in higher correlation between the low- and high-fidelity model response. We remark that the main advantages with respect to [41] are that we do not require a surrogate model to be computed beforehand, and that we replace normalizing flows with simple projections on the inverse cumulative distribution function, which benefits from increased robustness. Furthermore, we provided a complete theoretical analysis of this approach under idealized conditions, where we prove that the use of NeurAM results in higher correlation.

We presented extensive numerical experimentation that showed the advantages of this nonlinear technique for dimensionality reduction, and corroborated the analysis. We first considered simple two-dimensional models, and then moved to more complex test cases, including the Hartmann problem, and a cardiac electrophysiology model that presents a bifurcation in the parameter space. NeurAM produces smaller errors compared to linear approaches such as AS [12], or more recent nonlinear approaches like AM [7]. On the other hand, our method has a slightly larger variability which is mainly caused by the fact that the solution to the minimization problem that determines the NeurAM is not unique. Therefore, even if the error is in most of the cases smaller than for traditional approaches in the literature, we are interested in looking for additional conditions to enforce, in order to guarantee the uniqueness of the NeurAM. Moreover, the theoretical argument in Section 3.2.1 is valid under idealized conditions. An interesting future development consists in lifting this assumption, considering a more concrete scenario. Finally, we treated the bifurcation in the cardiac electrophysiology model by first performing a classification, and then computing NeurAM separately for the two classes. In future work, we would like the algorithm to be able to autonomously perform domain decomposition, without the need of a pre-trained classifier. In this context, we might benefit from a two-dimensional manifold, and therefore we should first study an extension to NeurAM with multidimensional latent space.

Acknowledgements

This work is supported by NSF CAREER award #1942662 (DES), NSF CDS&E award #2104831 (DES), NSF award #2105345 (ALM), and NIH grants #R01EB029362 and #R01HL167516 (ALM). This work used computational resources from the Stanford Research Computing Center (SRCC). Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan. G. Geraci was partially supported by a Laboratory Directed Research & Development (LDRD) project.

References

- [1] T. AKIBA, S. SANO, T. YANASE, T. OHTA, AND M. KOYAMA, *Optuna: A next-generation hyperparameter optimization framework*, 2019.
- [2] P. BALDI, *Autoencoders, unsupervised learning, and deep architectures*, in Proceedings of ICML Workshop on Unsupervised and Transfer Learning, I. Guyon, G. Dror,

- V. Lemaire, G. Taylor, and D. Silver, eds., vol. 27 of *Proceedings of Machine Learning Research*, Bellevue, Washington, USA, 02 Jul 2012, PMLR, pp. 37–49.
- [3] D. C. BARTOS, E. GRANDI, AND C. M. RIPPLINGER, *Ion Channels in the Heart*, 2015, pp. 1423–1464.
 - [4] G. BOMARITO, P. LESER, J. WARNER, AND W. LESER, *On the optimization of approximate control variates with parametrically defined estimators*, *Journal of Computational Physics*, 451 (2022), p. 110882.
 - [5] G. BONCORAGLIO AND C. FARHAT, *Active manifold and model-order reduction to accelerate multidisciplinary analysis and optimization*, *AIAA Journal*, 59(11) (2021).
 - [6] T. BRENNAN, M. FINK, AND B. RODRIGUEZ, *Multiscale modelling of drug-induced effects on cardiac electrophysiological activity*, *European Journal of Pharmaceutical Sciences*, 36 (2009), pp. 62–77.
 - [7] R. A. BRIDGES, A. D. GRUBER, C. FELDER, M. E. VERMA, AND C. HOFF, *Active manifolds: A non-linear analogue to active subspaces*, in *International Conference on Machine Learning*, 2019.
 - [8] C. J. C. BURGESS, *Dimension reduction: A guided tour*, *Foundations and Trends® in Machine Learning*, 2 (2010), pp. 275–365.
 - [9] P. COLLI FRANZONE, L. PAVARINO, AND S. SCACCHI, *Mathematical cardiac electrophysiology*, vol. 13, Springer, 2014.
 - [10] P. G. CONSTANTINE, *Active Subspaces - Emerging Ideas for Dimension Reduction in Parameter Studies*, 2015.
 - [11] P. G. CONSTANTINE AND P. DIAZ, *Global sensitivity metrics from active subspaces*, *Reliability Engineering & System Safety*, 162 (2017), pp. 1–13.
 - [12] P. G. CONSTANTINE, E. DOW, AND Q. WANG, *Active subspace methods in theory and practice: Applications to kriging surfaces*, *SIAM Journal on Scientific Computing*, 36 (2014), pp. A1500–A1524.
 - [13] P. G. CONSTANTINE, C. KENT, AND T. BUI-THANH, *Accelerating Markov chain Monte Carlo with active subspaces*, *SIAM J. Sci. Comput.*, 38 (2016), pp. A2779–A2805.
 - [14] N. DUAN AND K.-C. LI, *Slicing Regression: A Link-Free Regression Method*, *The Annals of Statistics*, 19 (1991), pp. 505 – 530.
 - [15] M. FEDELE, R. PIERSANTI, F. REGAZZONI, M. SALVADOR, P. C. AFRICA, M. BUCELLI, A. ZINGARO, L. DEDE’, AND A. QUARTERONI, *A comprehensive and biophysically detailed computational model of the whole human heart electromechanics*, *Computer Methods in Applied Mechanics and Engineering*, 410 (2023), p. 115983.
 - [16] G. GERACI, M. ELDRED, A. GORODETSKY, AND J. JAKEMAN, *Leveraging active directions for efficient multifidelity uncertainty quantification*, in *6th European Conference on Computational Mechanics (ECCM 6)*, 2018, pp. 2735–2746.
 - [17] A. GLAWS, P. G. CONSTANTINE, J. N. SHADID, AND T. M. WILDEY, *Dimension reduction in magnetohydrodynamics power generation models: Dimensional analysis and active subspaces*, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10 (2017), pp. 312–325.

- [18] I. GOODFELLOW, Y. BENGIO, AND A. COURVILLE, *Deep learning*, Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2016.
- [19] A. A. GORODETSKY, G. GERACI, M. S. ELDRED, AND J. D. JAKEMAN, *A generalized approximate control variate framework for multifidelity uncertainty quantification*, J. Comput. Phys., 408 (2020), pp. 109257, 29.
- [20] I. KOBYZEV, S. D. PRINCE, AND M. A. BRUBAKER, *Normalizing flows: An introduction and review of current methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 43 (2021), pp. 3964–3979.
- [21] K. KONTOLATI, D. LOUKREZIS, D. G. GIOVANIS, L. VANDANAPU, AND M. D. SHIELDS, *A survey of unsupervised learning methods for high-dimensional uncertainty quantification in black-box-type problems*, Journal of Computational Physics, 464 (2022), p. 111313.
- [22] S. KRISHNAMOORTHY, M. SARKAR, AND W. S. KLUG, *Numerical quadrature and operator splitting in finite element methods for cardiac electrophysiology*, Int. J. Numer. Methods Biomed. Eng., 29 (2013), pp. 1243–1266.
- [23] R. R. LAM, O. ZAHM, Y. M. MARZOUK, AND K. E. WILLCOX, *Multifidelity dimension reduction via active subspaces*, SIAM J. Sci. Comput., 42, pp. A929–A956.
- [24] O. P. LE MAITRE AND O. KNIO, *Spectral Methods for Uncertainty Quantification. With Applications to Computational Fluid Dynamics*, Springer Dordrecht, 2010.
- [25] K.-C. LI, *Sliced inverse regression for dimension reduction*, Journal of the American Statistical Association, 86 (1991), pp. 316–327.
- [26] E. LINDNER, A. MICHELETTI, AND C. NUNES, *Mathematical Modelling in Real Life Problems—Case Studies from ECMI-Modelling Weeks*, Mathematics in Industry, Springer, Cham, [2020] ©2020.
- [27] M. D. MCKAY, R. J. BECKMAN, AND W. J. CONOVER, *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code*, Technometrics, 42 (2000), pp. 55–61.
- [28] L. W. T. NG AND K. E. WILLCOX, *Multifidelity approaches for optimization under uncertainty*, Internat. J. Numer. Methods Engrg., 100 (2014), pp. 746–772.
- [29] A. PALACIOS, *Mathematical modeling—a dynamical systems approach to analyze practical problems in STEM disciplines*, Mathematical Engineering, Springer, Cham, [2022] ©2022.
- [30] B. PEHERSTORFER, K. WILLCOX, AND M. GUNZBURGER, *Optimal model management for multifidelity Monte Carlo estimation*, SIAM J. Sci. Comput., 38 (2016), pp. A3163–A3194.
- [31] R. PIERSANTI, F. REGAZZONI, M. SALVADOR, A. F. CORNO, L. DEDE’, C. VERGARA, AND A. QUARTERONI, *3D-0D closed-loop model for the simulation of cardiac biventricular electromechanics*, Comput. Methods Appl. Mech. Engrg., 391 (2022), pp. Paper No. 114607, 29.
- [32] E. PLISCHKE, E. BORGONOVO, AND C. L. SMITH, *Global sensitivity measures from given data*, European J. Oper. Res., 226 (2013), pp. 536–550.

- [33] A. QUARTERONI, L. DEDE', A. MANZONI, AND C. VERGARA, *Mathematical Modelling of the Human Cardiovascular System: Data, Numerical Approximation, Clinical Applications*, Cambridge University Press, 2019.
- [34] F. REGAZZONI, M. SALVADOR, P. C. AFRICA, M. FEDELE, L. DEDE', AND A. QUARTERONI, *A cardiac electromechanical model coupled with a lumped-parameter model for closed-loop blood circulation*, Journal of Computational Physics, 457 (2022), p. 111083.
- [35] T. M. RUSSI, *Uncertainty quantification with experimental data and complex system models*, 2010. PhD thesis.
- [36] M. SALVADOR, M. FEDELE, P. C. AFRICA, E. SUNG, L. DEDE', A. PRAKOSA, J. CHRISPIN, N. TRAYANOVA, AND A. QUARTERONI, *Electromechanical modeling of human ventricles with ischemic cardiomyopathy: numerical simulations in sinus rhythm and under arrhythmia*, Computers in Biology and Medicine, 136 (2021), p. 104674.
- [37] J. N. SHADID, R. P. PAWLOWSKI, E. C. CYR, R. S. TUMINARO, L. CHACÓN, AND P. D. WEBER, *Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton-Krylov-AMG*, Comput. Methods Appl. Mech. Engrg., 304 (2016), pp. 1–25.
- [38] I. M. SOBOL, *Estimation of the sensitivity of nonlinear mathematical models*, Mat. Model., 2 (1990), pp. 112–118.
- [39] K. H. TEN TUSSCHER AND A. V. PANFILOV, *Alternans and spiral breakup in a human ventricular tissue model*, American Journal of Physiology. Heart and Circulatory Physiology, 291 (2006), pp. 1088–1100.
- [40] R. TIPIREDDY AND R. GHANEM, *Basis adaptation in homogeneous chaos spaces*, Journal of Computational Physics, 259 (2014), pp. 304–317.
- [41] A. ZANONI, G. GERACI, M. SALVADOR, K. MENON, A. L. MARSDEN, AND D. E. SCHIAVAZZI, *Improved multifidelity Monte Carlo estimators based on normalizing flows and dimensionality reduction techniques*, Comput. Methods Appl. Mech. Engrg., 429 (2024), p. Paper No. 117119.
- [42] A. ZANONI, G. GERACI, M. SALVADOR, K. MENON, A. L. MARSDEN, AND D. E. SCHIAVAZZI, *Linear and nonlinear dimension reduction strategies for multifidelity uncertainty propagation of nonparametric distributions*, AIAA SciTech Forum, (2024).
- [43] X. ZENG, G. GERACI, M. S. ELDRED, J. D. JAKEMAN, A. A. GORODETSKY, AND R. GHANEM, *Multifidelity uncertainty quantification with models based on dissimilar parameters*, Comput. Methods Appl. Mech. Engrg., 415 (2023), pp. Paper No. 116205, 36.
- [44] X. ZENG, G. GERACI, A. GORODETSKY, J. JAKEMAN, M. S. ELDRED, AND R. GHANEM, *Improving bayesian networks multifidelity surrogate construction with basis adaptation*, in Proceedings of the AIAA Forum 2023, AIAA 2023-0917, 2023.
- [45] X. ZENG, J. RED-HORSE, AND R. GHANEM, *Accelerated basis adaptation in homogeneous chaos spaces*, Computer Methods in Applied Mechanics and Engineering, 386 (2021), p. 114109.