SAND2024-07517J

Improved multifidelity Monte Carlo estimators based on normalizing flows and dimensionality reduction techniques

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

Abstract

We study the problem of multifidelity uncertainty propagation for computationally expensive models. In particular, we consider the general setting where the high-fidelity and low-fidelity models have a dissimilar parameterization both in terms of number of random inputs and their probability distributions, which can be either known in closed form or provided through samples. We derive novel multifidelity Monte Carlo estimators which rely on a shared subspace between the high-fidelity and low-fidelity models where the parameters follow the same probability distribution, i.e., a standard Gaussian. We build the shared space employing normalizing flows to map different probability distributions into a common one, together with linear and nonlinear dimensionality reduction techniques, active subspaces and autoencoders, respectively, which capture the subspaces where the models vary the most. We then compose the existing low-fidelity model with these transformations and construct modified models with an increased correlation with the high-fidelity model, which therefore yield multifidelity estimators with reduced variance. A series of numerical experiments illustrate the properties and advantages of our approaches.

Keywords. multifidelity, uncertainty quantification, Monte Carlo estimators, active subspaces, autoencoders, normalizing flows.

1 Introduction

Uncertainty quantification has become a crucial component of computational modeling, as a way to enhance the validity and utility of numerical simulations. Uncertainty quantification studies can provide confidence metrics for quantities of interest and inform future data collection through sensitivity and identifiability analysis. However, in many cases, a naive approach to uncertainty quantification quickly becomes computationally infeasible as a result of the large computational cost needed to numerically solve complex physics-based mathematical models. Therefore, maintaining a reasonable computational cost for an uncertainty quantification study becomes challenging when relying solely on high-fidelity

[*]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.

simulations. This has motivated the development of multilevel and multifidelity Monte Carlo strategies that offset the computational cost of estimation to low-fidelity models, accelerating convergence and improving the tractability of uncertainty quantification for computationally expensive simulations [3, 6, 14, 15, 27, 29, 34, 35].

In some cases, however, the low fidelity-model can be obtained through a substantial simplification of the high-fidelity model, resulting in input parameters with potentially both different dimensionality and probability distribution, which we refer to as dissimilar parameterization. We remark that in this case the performance of standard multifidelity Monte Carlo estimators decreases, and it is important to concentrate the variability in few dimensions to enhance the correlation [12, 45]. We therefore propose two methodologies to create a shared subspace of reduced dimension which acts as a bridge between the two models. The first step in obtaining the shared subspace consists of finding an "important" subspace individually for both the high-fidelity and low-fidelity models where they vary the most, and consequently which captures most of the variance of the models. Perturbations of the random inputs along such important directions are responsible for most of the variability in the model response. If such a structure is present in the problem, then we can consider model responses only for inputs in the subspace, effectively reducing the problem dimensionality. In this work we employ two different dimensionality reduction techniques, active subspaces and autoencoders, which provide linear and nonlinear transformations, respectively. Active subspaces have been first formalized in [4, 5], while an introduction about autoencoder for unsupervised learning can be found in [2]. Autoencoders have become popular in the past few years thanks to their expressibility, as they leverage neural networks to find nonlinear transformations of the data, as opposed to linear maps provided by active subspaces. The second ingredient to obtain a shared subspace is a normalizing flow, i.e., an invertible transformation from a generic probability distribution into an easy-to-sample base distribution, usually a standard Gaussian. For a comprehensive review about normalizing flows we refer to [22, 28]. The goal of normalizing flows is enforcing the same probability distribution, in particular a standard Gaussian, of the latent variables, i.e., the parameters in the shared space. For active subspaces, which are known to preserve Gaussianity, we build normalizing flows from the input distributions of the parameters, such that the reduced subspaces of both the high-fidelity and low-fidelity models are automatically Gaussian. On the other hand, for autoencoders we learn normalizing flows from the distributions of the latent variables of the model into the shared subspace, which is therefore Gaussian. We remark that these subspaces and normalizing flows are approximated from the results of a pilot run that is typically employed as a first step in any multifidelity estimator, without the need to run additional high-fidelity simulations.

We reasonably assume that if the variability of the high-fidelity and low-fidelity models is concentrated in few variables, then using the shared space as a bridge aligning the important directions of the fidelities would improve the correlations and consequently reduce the variance of the multifidelity estimator. Starting from the existing low-fidelity model, we therefore construct new low-fidelity models whose inputs are the same parameters of the high-fidelity model which are transformed into inputs for the original low-fidelity model in such a way that the correlation between the fidelities increases. We finally obtain multifidelity estimators which are unbiased, as the high-fidelity model does not change, and with reduced variance with respect to standard multifidelity Monte Carlo estimators. The present work generalizes and extends a series of contributions in this area like, e.g., [11, 12, 45], where the challenge of dissimilar parameterization has been tackled with either active subspaces or adaptive basis [42]. The main extension regarding

previous work on active subspaces is that we include a normalizing flow which allows for any input distribution of the parameters, even known only through samples. Moreover, both active subspaces and adaptive basis are linear dimension reduction strategies, and therefore autoencoders are a natural extension since they provide nonlinear transformations.

We apply our methodology to challenging examples, such as reaction-diffusion equations with applications in biological pattern formation and cardiovascular simulations for a coronary model with stenosis. Uncertainty quantification has recently gained momentum in the field of cardiovascular modeling, with recent works exploring a range of methods which can be used to address various sources of uncertainty within these models [8,9,33,36,37,39]. Moreover, various simplifying assumptions can be made to cardiovascular hemodynamics to generate low-fidelity models of intermediate complexity for multifidelity uncertainty propagation. In fact, integrating the Navier–Stokes equations on the vessel cross sections leads to one-dimensional hemodynamic models [18], and a linearization of the incompressible Navier–Stokes equations around rest conditions leads to an even simpler zero-dimensional formulation utilizing analogous electrical circuits to solve vascular networks [31,32]. These low fidelity models grant us multiple orders of magnitude cost savings over a full three-dimensional model.

Outline. The rest of the paper is organized as follows. In Section 2 we give background on multifidelity uncertainty quantification, active subspaces, autoencoders, and normalizing flows, which we employ in the definition of our novel methods, which are described in details in Section 3. Then, in Section 4 we present several numerical experiments, and we finally draw conclusions in Section 5.

2 Review of current methods

In this section we briefly review the main tools employed in our uncertainty quantification pipelines. We recall that our goal is the efficient estimation of scalar quantities of interest of computationally expensive models. Let $Q\colon \mathbb{R}^d \to \mathbb{R}$ represent a computational model, and let $\boldsymbol{\xi} \in \mathbb{R}^d$ be a random vector of inputs distributed according to the joint distribution μ. We aim to characterize the statistical moments of the quantity of interest $Q(\boldsymbol{\xi})$, focusing in particular on its expectation $\mathbb{E}[Q(\boldsymbol{\xi})]$. For higher-order moments, we can just replace the quantity of interest $Q(\boldsymbol{\xi})$ by its power $Q^m(\boldsymbol{\xi})$ with $m > 1$. The standard Monte Carlo estimator approximates this expected value through a set of N realizations $\{\boldsymbol{\xi}_n\}_{n=1}^N$ of the input variable $\boldsymbol{\xi} \sim \mu$ as

$$\widehat{Q}_N^{\mathrm{MC}} = \frac{1}{N} \sum_{n=1}^{N} Q(\boldsymbol{\xi}_n).$$

This estimator is simple to compute and it is unbiased, in the sense that

$$\mathbb{E}\left[\widehat{Q}_N^{\mathrm{MC}}\right] = \mathbb{E}[Q(\boldsymbol{\xi})].$$

However, its root mean squared error is $\mathcal{O}(N^{-1/2})$, meaning that a large number of evaluations might be necessary in order to reach the desired accuracy. In concrete applications where the evaluation of Q is computationally expensive, increasing the number of samples N can be intractable. Therefore, multifidelity Monte Carlo estimators have been developed to overcome this issue.

2.1 Multifidelity Monte Carlo estimator

Let us assume that a computationally cheap model for the original model Q is available, and denote Q^{HF} and Q^{LF} the original (high-fidelity) and the cheap (low-fidelity) models, respectively. We notice that the low-fidelity model can be any, even biased, approximation of the high-fidelity model, as long as it is computationally cheap to simulate. We adopt the estimator originally introduced in [27], which focus on the case of a single low-fidelity model, but the multifidelity Monte Carlo estimator can be easily extended, or generalized, to the case of multiple low-fidelity models, e.g., [3, 6, 15, 29, 34, 35]. Let $w = \mathcal{C}^{\mathrm{LF}}/\mathcal{C}^{\mathrm{HF}}$ be the cost ratio between the two fidelities, and let $\mathcal{B}$ be the computational budget available in terms of high-fidelity evaluations, i.e.,

$$\mathcal{B} = N^{\mathrm{HF}} + wN^{\mathrm{LF}},$$

where N^{HF} and N^{LF} are the numbers of high-fidelity and low-fidelity evaluations, respectively. We aim to split the computationally budget between the high-fidelity and low-fidelity models in such a way that the final multifidelity estimator has the smallest possible variance [29]. Assuming that our budget is large enough, this is achieved by setting

$$N^{\mathrm{HF}} = \frac{\mathcal{B}}{1+w\gamma} \qquad \text{and} \qquad N^{\mathrm{LF}} = \gamma N^{\mathrm{HF}} = \frac{\gamma\mathcal{B}}{1+w\gamma}, \qquad \text{with} \qquad \gamma = \sqrt{\frac{\rho^2}{w(1-\rho^2)}}, \tag{2.1}$$

where ρ is the Pearson correlation coefficient between the HF and LF models

$$\rho = \frac{\mathbb{C}\mathrm{ov}\left(Q^{\mathrm{HF}}(\boldsymbol{\xi}), Q^{\mathrm{LF}}(\boldsymbol{\xi})\right)}{\sqrt{\mathbb{V}\mathrm{ar}[Q^{\mathrm{HF}}(\boldsymbol{\xi})]\,\mathbb{V}\mathrm{ar}\left[Q^{\mathrm{LF}}(\boldsymbol{\xi})\right]}}.$$

Once the number of evaluations for each model has been selected, the multifidelity Monte Carlo estimator (MFMC) is defined as

$$\begin{aligned}\widehat{Q}^{\mathrm{MFMC}}_{N^{\mathrm{HF}},N^{\mathrm{LF}}} &= \widehat{Q}^{\mathrm{HF,MC}}_{N^{\mathrm{HF}}} - \beta\left(\widehat{Q}^{\mathrm{LF,MC}}_{N^{\mathrm{HF}}} - \widehat{Q}^{\mathrm{LF,MC}}_{N^{\mathrm{LF}}}\right) \\ &= \frac{1}{N^{\mathrm{HF}}}\sum_{n=1}^{N^{\mathrm{HF}}} Q^{\mathrm{HF}}(\boldsymbol{\xi}_n) - \beta\left(\frac{1}{N^{\mathrm{HF}}}\sum_{n=1}^{N^{\mathrm{HF}}} Q^{\mathrm{LF}}(\boldsymbol{\xi}_n) - \frac{1}{N^{\mathrm{LF}}}\sum_{n=1}^{N^{\mathrm{LF}}} Q^{\mathrm{LF}}(\boldsymbol{\xi}_n)\right),\end{aligned}$$

where the optimal value for the coefficient β is given by

$$\beta = \frac{\mathbb{C}\mathrm{ov}\left(Q^{\mathrm{HF}}(\boldsymbol{\xi}), Q^{\mathrm{LF}}(\boldsymbol{\xi})\right)}{\mathbb{V}\mathrm{ar}\left[Q^{\mathrm{LF}}(\boldsymbol{\xi})\right]}, \tag{2.2}$$

and N^{HF} samples are shared between the two models.

Remark 2.1. Computing the optimal values for the numbers N^{HF} and N^{LF} of evaluations and the coefficient β in equations (2.1) and (2.2), respectively, is important to take full advantage of the method and get the smallest possible variance for the multifidelity Monte Carlo estimator. However, it is not essential to employ the optimal values for $N^{\mathrm{HF}}, N^{\mathrm{LF}}$, and β, and any choice would still produce an unbiased estimator. In case the optimal values are used, then we obtain

$$\mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{MFMC}}_{N^{\mathrm{HF}},N^{\mathrm{LF}}}\right] = \mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{HF,MC}}_{\mathcal{B}}\right]\left(\sqrt{1-\rho^2} + \sqrt{w\rho^2}\right)^2, \tag{2.3}$$

which yields that

$$|\rho| > \frac{4w}{(1+w)^2} \qquad \Longrightarrow \qquad \mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{MFMC}}_{N^{\mathrm{HF}},N^{\mathrm{LF}}}\right] < \mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{HF,MC}}_{\mathcal{B}}\right].$$

Therefore, variance reduction with respect to standard Monte Carlo is guaranteed as long as the HF and LF models are well correlated.

This estimator, as with others available in literature, can be obtained as instances of the so-called Approximate Control Variate (ACV) approach introduced in [15]. As a consequence, the methodologies developed here will be applicable to a larger set of estimators. In the presence of dissimilar parameterization, retaining high correlation among models is paramount for the efficiency of the estimator. In the next section we introduce active subspaces as a way to increase the correlation between the high-fidelity and low-fidelity models. Moreover, active subspaces act as a bridge between models having a different number of inputs.

2.2 Active subspaces

The active subspaces approach is a methodology which is usually employed in uncertainty quantification studies in order to reduce the dimension of the random inputs, without sacrificing accuracy in approximating a quantity of interest [5]. It allows one to find the dominant directions, i.e., the linear subspace where the quantity of interest Q varies the most. Let C be the matrix which quantifies the variation defined as

$$C = \mathbb{E}\left[\nabla Q(\boldsymbol{\xi})\nabla Q(\boldsymbol{\xi})^\top\right], \tag{2.4}$$

where the expectation is taken with respect to the measure μ and the gradient is computed with respect to the variable $\boldsymbol{\xi}$, and which can be approximated using a collection of samples $\{\boldsymbol{\xi}_n\}_{n=1}^N$ as

$$C \simeq \widetilde{C} = \frac{1}{N}\sum_{n=1}^{N} \nabla Q(\boldsymbol{\xi}_n)\nabla Q(\boldsymbol{\xi}_n)^\top.$$

We remark that the gradient of the model Q is required in order to compute the matrix C. It can be approximated through finite differences, linear approximations, surrogate models, or any other technique to compute derivatives numerically. Additional details about how we deal with the gradient in our work are given in Remark 3.1. Moreover, we note that more sophisticated strategies could be introduced to reduce the data requirment of this step, especially for the high-fidelity model. For instance, the approximation of C could be obtained via a multifidelity approach as demonstrated in [23]. Note that since $\widetilde{C}$ is a symmetric positive semidefinite matrix, then it admits a real eigenvalue decomposition

$$\widetilde{C} = W\Lambda W^\top \qquad \text{with} \qquad W, \Lambda \in \mathbb{R}^{d\times d},$$

where W is orthogonal and its columns are the eigenvectors of C, and Λ is a diagonal matrix which contains the corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$, which can be arranged in decreasing order. This ordering suggests a possible separation between important (or active) and irrelevant (or inactive) contributions of the corresponding eigenvector to the linear decomposition of C as a sum of rank-one matrices, in particular if the smallest eigenvalues are close to zero. To better visualize such separation, we write

$$\Lambda = \begin{bmatrix} \Lambda_A & \\ & \Lambda_I \end{bmatrix} \qquad \text{and} \qquad W = \begin{bmatrix} W_A & W_I \end{bmatrix},$$

where $\Lambda_A \in \mathbb{R}^{r\times r}$ and $W_A \in \mathbb{R}^{d\times r}$ contain the first $r < d$ eigenvalues and eigenvectors, respectively. The column spans of W_A and W_I represent the active and inactive subspace, respectively. These linear transformations allow us to decompose the original input $\boldsymbol{\xi} \in \mathbb{R}^d$ into the active and inactive parts as follows

$$\boldsymbol{\xi} = W_A\boldsymbol{\xi}_A + W_I\boldsymbol{\xi}_I, \qquad \text{where} \qquad \boldsymbol{\xi}_A = W_A^\top\boldsymbol{\xi} \in \mathbb{R}^r, \quad \xi_I = W_I^\top\boldsymbol{\xi} \in \mathbb{R}^{d-r}. \tag{2.5}$$

If the dimensionality of the problem can actually be reduced, then the inactive component of the decomposition can be ignored because it gives a negligible contribution to the quantity of interest, i.e., $Q(\boldsymbol{\xi}) \simeq Q(W_A\boldsymbol{\xi}_A)$, and the problem becomes r-dimensional.

Remark 2.2. The probability distribution of the random inputs μ often results from a modeling choice, particularly when observational data are insufficient, and the matrix C in equation (2.4) is dependent on this distribution. Therefore, we notice that different choices for μ lead to different active subspaces for the same model.

The distribution of the active part in the decomposition (2.5) is in general unknown. Nevertheless, if the distribution of the data μ is the standard Gaussian, i.e., $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, I_d)$, then the active component remains standard Gaussian, i.e., $\boldsymbol{\xi}_A \sim \mathcal{N}(\mathbf{0}, I_r)$, and in fact a linear transformation of a Gaussian random variable is still Gaussian, and it holds

$$\mathbb{E}[\boldsymbol{\xi}_A] = W_A^\top\,\mathbb{E}[\boldsymbol{\xi}] = 0 \qquad \text{and} \qquad \mathbb{V}\text{ar}[\boldsymbol{\xi}_A] = W_A^\top\,\mathbb{V}\text{ar}[\boldsymbol{\xi}]W_A = W_A^\top W_A = I_r.$$

This suggests the possibility of generating shared parameterizations even for models having a different number of random inputs, as long as their active inputs share the same dimensionality and distribution [11, 12]. For general probability distributions, it is often possible to define a transformation mapping μ to a standard Gaussian via normalizing flows, compute the active subspace, and finally transform the variables back to their original distribution after sampling. We notice that this is a significant advantage since it allows one to handle any type of input data, independently of their correlation. As observed in [11], even if a transformation introduces additional complexity, the increase in computational cost could be outweighed by the increase in correlation between models. A strong limitation of the active subspace technique is that it provides only linear maps for dimensionality reduction. This restriction can be overcome by replacing active subspaces with autoencoders, which allow for nonlinear transformations. The next two sections will therefore focus on autoencoders and normalizing flows, respectively.

2.3 Autoencoders

Autoencoders are a data-driven approach widely used for unsupervised dimensionality reduction, with the ability to learn an intrinsic structure existing in the data by leveraging neural networks. They consist of one encoder $\mathcal{E}$ followed by one decoder $\mathcal{D}$, where the encoder $\mathcal{E}\colon \mathbb{R}^d \to \mathbb{R}^r$ with $r < d$ compresses the original input $\boldsymbol{\xi} \in \mathbb{R}^d$ into a latent representation $\boldsymbol{x} \in \mathbb{R}^r$, and the decoder $\mathcal{D}\colon \mathbb{R}^r \to \mathbb{R}^d$ seeks to reconstruct the original input $\boldsymbol{\xi}$ starting from the latent representation $\boldsymbol{x}$, in the sense that $\mathcal{D}(\mathcal{E}(\boldsymbol{\xi})) \simeq \boldsymbol{\xi}$. The functions $\mathcal{E}$ and $\mathcal{D}$ are usually parameterized by fully connected neural networks. In this work we are not interested in reconstructing exactly the original input, but rather $Q(\boldsymbol{\xi})$. In particular, given a model Q, analogously to the active subspace approach, we would like to find a lower dimensional representation of Q by selecting a manifold of dimension r where the function varies the most. Hence, we aim to reconstruct the original quantity of interest, meaning that $Q(\mathcal{D}(\mathcal{E}(\boldsymbol{\xi}))) \simeq Q(\boldsymbol{\xi})$. We notice that this can be seen as a supervised dimensionality

reduction, where we would like to learn a structure in the data according to some metric, which in this case is the model Q. We parameterize the encoder and the decoder as fully connected neural networks $\mathcal{E}(\cdot;\phi_E)$ and $\mathcal{D}(\cdot;\phi_D)$ with hyperbolic tangent activation functions, and we compute the optimal parameters by minimizing the loss function

$$\mathcal{L}_{\text{AE}}(\phi_E,\phi_D) = \frac{1}{N}\sum_{n=1}^{N} \left|Q(\boldsymbol{\xi}_n) - Q(\mathcal{D}(\mathcal{E}(\boldsymbol{\xi}_n;\phi_E);\phi_D))\right|,$$

where $\{\boldsymbol{\xi}_n\}_{n=1}^N$ is the sample of available data. We notice that minimizing the loss function requires multiple evaluations of the model Q, and this can be impractical if the model is computationally expensive. We therefore build a surrogate model which is only used in the training process, as we discuss in Remark 3.3. Using the same terminology as for the active subspaces technique, we say that the active variable is $\boldsymbol{\xi}_A = \mathcal{E}(\boldsymbol{\xi})$ and we notice that its distribution is unknown. Therefore, by constructing a map from the probability distribution of the active variable into a standard Gaussian through normalizing flows (see Section 2.4), we can generate a shared parameterization even for models having a different number of inputs, as long as the reduced dimension of the autoencoder is the same.

2.4 Normalizing flows

Normalizing flows are invertible transformations which map generic probability distributions into more simple and tractable distributions, usually standard Gaussian. Let μ be a probability distribution on $\mathbb{R}^d$ with density ψ and let μ_0 with density ψ_0 be the target distribution, which in our work, like in most cases, is $\mu_0 = \mathcal{N}(\mathbf{0}, I_d)$. We aim to find a diffeomorphism $\mathcal{T}\colon \mathbb{R}^d \to \mathbb{R}^d$ such that $\mathcal{T}_{\#}\mu = \mu_0$, where $\mathcal{T}_{\#}$ denotes the pushforward measure through the map $\mathcal{T}$. We consider a parameterization $\mathcal{T}(\cdot;\theta)$, which is the composition of invertible transformations defined by neural networks. Then, given a sample $\{\boldsymbol{\xi}_n\}_{n=1}^N$ from the initial distribution μ, the best parameters θ are computed by maximizing the log-likelihood function for the density $\widetilde{\psi}(\cdot;\theta)$ of the measure $\mathcal{T}^{-1}(\cdot;\theta)_{\#}\mu_0$. In practice, we minimize the loss function

$$\mathcal{L}_{\text{NF}}(\theta) = -\sum_{n=1}^{N} \log \widetilde{\psi}(\boldsymbol{\xi}_n;\theta),$$

which, by the change of variable formula, can be written as

$$\mathcal{L}_{\text{NF}}(\theta) = -\sum_{n=1}^{N} \left[\log \psi_0(\mathcal{T}(\boldsymbol{\xi}_n;\theta)) + \log\left|\det \nabla\mathcal{T}(\boldsymbol{\xi}_n;\theta)\right|\right]. \tag{2.6}$$

Remark 2.3. If the target distribution is standard Gaussian, i.e., $\mu_0 = \mathcal{N}(\mathbf{0}, I_d)$, then ψ_0 is the density of a multivariate normal distribution, and equation (2.6) reads

$$\mathcal{L}_{\text{NF}}(\theta) = \sum_{n=1}^{N} \left[\frac{1}{2}\left\|\mathcal{T}(\boldsymbol{\xi}_n;\theta)\right\|^2 - \log\left|\det \nabla\mathcal{T}(\boldsymbol{\xi}_n;\theta)\right|\right] + \frac{Nd}{2}\log(2\pi),$$

where the last term in the right-hand side can be neglected since it is independent of θ.

Therefore, a good parameterization for a normalizing flow needs to be sufficiently expressive, in order to be able to approximate the exact transformation, and efficient in terms of computation of the map itself, and the determinant of its Jacobian matrix. Moreover,

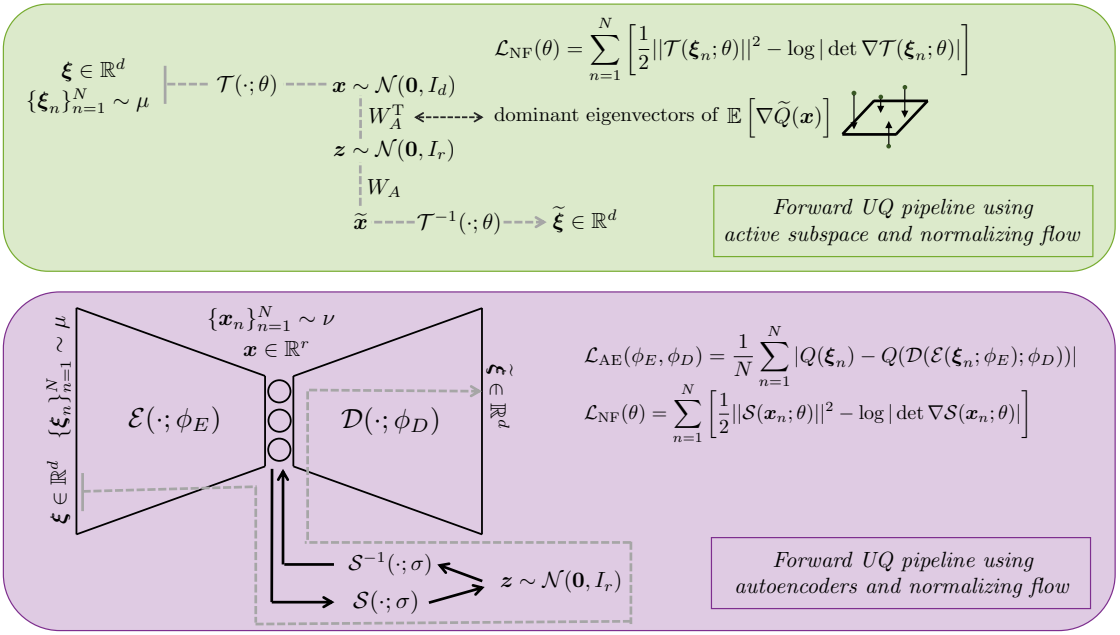

Figure 1: Summary of the methodologies presented in Section 3, which holds for both the high-fidelity and low-fidelity models.

the computation of the inverse of the map should also be cheap, so that we can draw new samples from the initial distribution μ efficiently. Indeed, a straightforward way to get a new sample from μ consists of drawing a sample $\boldsymbol{x} \sim \mu_0$ and then applying the inverse of the learned map, i.e., $\boldsymbol{\xi} = \mathcal{T}^{-1}(\boldsymbol{x};\theta)$. In this work we employ both the RealNVP normalizing flow, which is presented in [7] and implemented in [40], and splines from the FlowTorch package (https://www.flowtorch.ai). We also include a deterministic transformation (inverse of the hyperbolic tangent) to map a distribution with finite support, e.g., uniform, into a distribution with infinite support, e.g., standard Gaussian. We remark that normalizing flows have already been used in uncertainty quantification. Some examples are [24,38], where a specific type of map given by the the Knothe-Rosenblatt rearrangement is employed, and [44], where normalizing flows are combined with adaptive surrogate modeling.

3 Enhancing multifidelity estimator performance

We now describe our two methodologies for improving multifidelity Monte Carlo estimators. Our pipelines have the twofold purpose of reducing the variance of the estimator and increasing its range of applicability to problems where the original model and its low-fidelity version have a dissimilar parameterization. We recall that we consider an expensive high-fidelity model $Q^{\mathrm{HF}}\colon \mathbb{R}^{d^{\mathrm{HF}}} \to \mathbb{R}$ and a cheap low-fidelity model $Q^{\mathrm{LF}}\colon \mathbb{R}^{d^{\mathrm{LF}}} \to \mathbb{R}$, and we aim to estimate the expectation of the former model $\mathbb{E}[Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}})]$ under the assumption that the input parameters $\boldsymbol{\xi}^{\mathrm{HF}} \in \mathbb{R}^{d^{\mathrm{HF}}}$ and $\boldsymbol{\xi}^{\mathrm{LF}} \in \mathbb{R}^{d^{\mathrm{LF}}}$ are distributed according to some probability distribution μ^{HF} and μ^{LF}, respectively. We notice that, in the most general setting, we allow the number of input parameters $d^{\mathrm{HF}}, d^{\mathrm{LF}}$ and also their distributions $\mu^{\mathrm{HF}}, \mu^{\mathrm{LF}}$ to be different. Our goal is to create a shared parameterization of reduced dimension r, where the two models vary the most. We remark that we do not require the high-fidelity and low-fidelity models to naturally share the same lower dimensional manifold, but our main

goal is finding the lower dimensional subspace of each model separately, and then create a link between them through a shared subspace. We then assume that we want to keep the original high-fidelity model for two reasons. First, we could not have additional resources for generating new high-fidelity simulations on the shared space, and second we would like to preserve the unbiasedness of the multifidelity estimator. On the other hand, we create a modified low-fidelity model which is better correlated to the high-fidelity one. We achieve this by employing normalizing flows and dimensionality reduction techniques, in particular active subspaces for the method in Section 3.1 and autoencoders for the method in Section 3.2. We remark that the main difference between the two approaches is that autoencoders allow for nonlinear transformations which cannot be obtained from the active subspace technique. A summary of the two methodologies, which are outlined in the next two sections, is showed in Fig. 1.

3.1 Coupling MFMC with active subspaces and normalizing flows

In this section we create a shared parameterization between Q^{HF} and Q^{LF} using the active variables given by the active subspaces of the two models. Let $\{\boldsymbol{\xi}_n^{\mathrm{HF}}\}_{n=1}^N \sim \mu^{\mathrm{HF}}$ and $\{\boldsymbol{\xi}_n^{\mathrm{LF}}\}_{n=1}^N \sim \mu^{\mathrm{LF}}$ be samples from the input distributions, and consider the normalizing flows $\mathcal{T}^{\mathrm{HF}}(\cdot;\theta^{\mathrm{HF}})\colon \mathbb{R}^{d^{\mathrm{HF}}} \to \mathbb{R}^{d^{\mathrm{HF}}}$ and $\mathcal{T}^{\mathrm{LF}}(\cdot;\theta^{\mathrm{LF}})\colon \mathbb{R}^{d^{\mathrm{LF}}} \to \mathbb{R}^{d^{\mathrm{LF}}}$ such that

$$\mathcal{T}^{\mathrm{HF}}(\cdot;\theta^{\mathrm{HF}})_{\#}\mu^{\mathrm{HF}} = \mathcal{N}(\mathbf{0}, I_{d^{\mathrm{HF}}}) \qquad \text{and} \qquad \mathcal{T}^{\mathrm{LF}}(\cdot;\theta^{\mathrm{LF}})_{\#}\mu^{\mathrm{LF}} = \mathcal{N}(\mathbf{0}, I_{d^{\mathrm{LF}}}), \tag{3.1}$$

and which, due to Remark 2.3, are obtained minimizing the loss functions

$$\begin{aligned}
\mathcal{L}_{\mathrm{NF}}^{\mathrm{HF}}(\theta^{\mathrm{HF}}) &= \sum_{n=1}^{N} \left[\frac{1}{2} \left\| \mathcal{T}^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}};\theta^{\mathrm{HF}}) \right\|^2 - \log \left| \det \nabla \mathcal{T}^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}};\theta^{\mathrm{HF}}) \right| \right], \\
\mathcal{L}_{\mathrm{NF}}^{\mathrm{LF}}(\theta^{\mathrm{LF}}) &= \sum_{n=1}^{N} \left[\frac{1}{2} \left\| \mathcal{T}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{LF}};\theta^{\mathrm{LF}}) \right\|^2 - \log \left| \det \nabla \mathcal{T}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{LF}};\theta^{\mathrm{LF}}) \right| \right].
\end{aligned}$$

We can now define the modified models $\widetilde{Q}^{\mathrm{HF}}\colon \mathbb{R}^{d^{\mathrm{HF}}} \to \mathbb{R}$ and $\widetilde{Q}^{\mathrm{LF}}\colon \mathbb{R}^{d^{\mathrm{LF}}} \to \mathbb{R}$, whose input distributions are standard Gaussian, employing the inverse of the normalizing flows

$$\widetilde{Q}^{\mathrm{HF}}(\boldsymbol{x}^{\mathrm{HF}}) = Q^{\mathrm{HF}}((\mathcal{T}^{\mathrm{HF}})^{-1}(\boldsymbol{x}^{\mathrm{HF}};\theta^{\mathrm{HF}})) \quad \text{and} \quad \widetilde{Q}^{\mathrm{LF}}(\boldsymbol{x}^{\mathrm{LF}}) = Q^{\mathrm{LF}}((\mathcal{T}^{\mathrm{LF}})^{-1}(\boldsymbol{x}^{\mathrm{LF}};\theta^{\mathrm{LF}})). \tag{3.2}$$

Applying the procedure described in Section 2.2 to $\widetilde{Q}^{\mathrm{HF}}$ and $\widetilde{Q}^{\mathrm{LF}}$, we compute the matrices

$$\widetilde{C}^{\mathrm{HF}} = \frac{1}{N}\sum_{n=1}^{N} \nabla\widetilde{Q}^{\mathrm{HF}}(\boldsymbol{x}_n^{\mathrm{HF}}) \nabla\widetilde{Q}^{\mathrm{HF}}(\boldsymbol{x}_n^{\mathrm{HF}})^{\top} \quad \text{and} \quad \widetilde{C}^{\mathrm{LF}} = \frac{1}{N}\sum_{n=1}^{N} \nabla\widetilde{Q}^{\mathrm{LF}}(\boldsymbol{x}_n^{\mathrm{LF}}) \nabla\widetilde{Q}^{\mathrm{LF}}(\boldsymbol{x}_n^{\mathrm{LF}})^{\top}, \tag{3.3}$$

where $\boldsymbol{x}_n^{\mathrm{HF}} = \mathcal{T}^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}};\theta^{\mathrm{HF}})$ and $\boldsymbol{x}_n^{\mathrm{LF}} = \mathcal{T}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{LF}};\theta^{\mathrm{LF}})$, and we learn the active subspaces $W_A^{\mathrm{HF}} \in \mathbb{R}^{d^{\mathrm{HF}}\times r}$ and $W_A^{\mathrm{LF}} \in \mathbb{R}^{d^{\mathrm{LF}}\times r}$ of dimension r, which capture the directions of maximum change of the two models. Hence, the shared space is obtained applying first the normalizing flow and then the transpose of the active subspace matrix as follows

$$\begin{aligned}
\boldsymbol{z}_n^{\mathrm{HF}} &= (W_A^{\mathrm{HF}})^T \boldsymbol{x}_n^{\mathrm{HF}} = (W_A^{\mathrm{HF}})^T \mathcal{T}^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}};\theta^{\mathrm{HF}}) \in \mathbb{R}^r, \\
\boldsymbol{z}_n^{\mathrm{LF}} &= (W_A^{\mathrm{LF}})^T \boldsymbol{x}_n^{\mathrm{LF}} = (W_A^{\mathrm{LF}})^T \mathcal{T}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{LF}};\theta^{\mathrm{LF}}) \in \mathbb{R}^r,
\end{aligned}$$

where both $\boldsymbol{z}_n^{\mathrm{HF}}$ and $\boldsymbol{z}_n^{\mathrm{LF}}$ are distributed accordingly to a standard Gaussian $\mathcal{N}(\mathbf{0}, I_r)$. We recall here that $\boldsymbol{\xi}$ stands for the original input, $\boldsymbol{x}$ is the corresponding normally distributed

parameter, and $\boldsymbol{z}$ is the active variable in the shared space. We remark that, if the optimal reduced dimensions differ between the two fidelities, the value of r should be chosen in the range given by these two dimensions finding the best trade-off between capturing all the variability of the models and increasing their correlation. We recall that our goal is increasing the correlation between the high-fidelity, which we do not modify, and the low fidelity models. We therefore construct a new low-fidelity model in which we map the input parameters of the high-fidelity model into the input parameters of the original low-fidelity model by means of the shared space. In particular, we define $\mathcal{Q}_{\text{AS}}^{\text{LF}}\colon \mathbb{R}^{d^{\text{HF}}} \to \mathbb{R}$ as

$$\mathcal{Q}_{\text{AS}}^{\text{LF}}(\boldsymbol{\xi}^{\text{HF}}) = Q^{\text{LF}}((\mathcal{T}^{\text{LF}})^{-1}(W_A^{\text{LF}}(W_A^{\text{HF}})^\top \mathcal{T}^{\text{HF}}(\boldsymbol{\xi}^{\text{HF}}; \theta^{\text{HF}}), \theta^{\text{LF}})), \tag{3.4}$$

and we employ it to introduce a new multifidelity estimator

$$\widehat{Q}_{N^{\text{HF}},N^{\text{LF}}}^{\text{MFMC,AS}} = \frac{1}{N^{\text{HF}}} \sum_{n=1}^{N^{\text{HF}}} Q^{\text{HF}}(\boldsymbol{\xi}_n^{\text{HF}}) - \beta \left(\frac{1}{N^{\text{HF}}} \sum_{n=1}^{N^{\text{HF}}} \mathcal{Q}_{\text{AS}}^{\text{LF}}(\boldsymbol{\xi}_n^{\text{HF}}) - \frac{1}{N^{\text{LF}}} \sum_{n=1}^{N^{\text{LF}}} \mathcal{Q}_{\text{AS}}^{\text{LF}}(\boldsymbol{\xi}_n^{\text{HF}}) \right). \tag{3.5}$$

We remark that since the high-fidelity model does not change, then $\widehat{Q}_{N^{\text{HF}},N^{\text{LF}}}^{\text{MFMC,AS}}$ is unbiased. In Algorithm 1 we summarize the main steps needed to construct the estimator. The same considerations presented in Section 2.1 regarding the choice of the coefficient β and of the numbers N^{HF} and N^{LF} of high-fidelity and low-fidelity evaluations are still valid here. Moreover, we remark that, if we do not have the possibility to generate new high-fidelity simulations, we can fix $N^{\text{HF}} = N$, where N is the dimension of the pilot sample used to build the new low-fidelity model, and increase only the number N^{LF} of low-fidelity simulations. In this case, we would not get the optimal variance reduction outlined in Remark 2.1, but we can still obtain a reduction in the variance of the multifidelity estimator due to the higher correlation between Q^{HF} and $\mathcal{Q}_{\text{AS}}^{\text{LF}}$ with respect to Q^{HF} and Q^{LF}.

Algorithm 1: MFMC AS

Input: High fidelity and low fidelity models Q^{HF} and Q^{LF}
Distributions μ^{HF} and μ^{LF} for the input parameters
or samples $\{\boldsymbol{\xi}_n^{\text{HF}}\}_{n=1}^N \sim \mu^{\text{HF}}$ and $\{\boldsymbol{\xi}_n^{\text{LF}}\}_{n=1}^N \sim \mu^{\text{LF}}$ from them
Computational budget $\mathcal{B}$.

Output: Estimation $\widehat{Q}_{N^{\text{HF}},N^{\text{LF}}}^{\text{MFMC,AS}}$ of $\mathbb{E}[Q^{\text{HF}}(\boldsymbol{\xi}^{\text{HF}})]$.

1: Compute the normalizing flows $\mathcal{T}^{\text{HF}}(\cdot;\theta^{\text{HF}})$ and $\mathcal{T}^{\text{LF}}(\cdot;\theta^{\text{LF}})$ which satisfy (3.1).
2: Compute the active subspaces W_A^{HF} and W_A^{LF} from the matrices $\widetilde{C}^{\text{HF}}$ and $\widetilde{C}^{\text{LF}}$ in (3.3) obtained from the modified models $\widetilde{Q}^{\text{HF}}$ and $\widetilde{Q}^{\text{LF}}$ in (3.2).
3: Define the new low-fidelity model $\mathcal{Q}_{\text{AS}}^{\text{LF}}$ in (3.4).
4: Compute the optimal allocation $N^{\text{HF}}, N^{\text{LF}}$ from a pilot sample.
5: Compute the estimator $\widehat{Q}_{N^{\text{HF}},N^{\text{LF}}}^{\text{MFMC,AS}}$ in (3.5).

Remark 3.1. In order to learn the active subspace matrices, it is necessary to compute the gradients of the models in equation (3.3), which might not be available or might be too computationally expensive. An approximation of the gradients can be obtained replacing the partial derivatives with finite differences, but this approach can be unfeasible, in particular

in high dimensions, even for the low-fidelity model. Therefore, we propose to train surrogate models $Q^{\mathrm{HF}}_{\mathrm{NN}}$ and $Q^{\mathrm{LF}}_{\mathrm{NN}}$ for Q^{HF} and Q^{LF} based on fully connected neural networks with ReLU activation functions, and then compute the gradients using automatic differentiation. Given a set of realizations $\{(\boldsymbol{\xi}^{\mathrm{HF}}_n, Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}_n))\}_{n=1}^N$ and $\{(\boldsymbol{\xi}^{\mathrm{LF}}_n, Q^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}_n))\}_{n=1}^N$, we obtain the neural networks $Q^{\mathrm{HF}}_{\mathrm{NN}}(\cdot;\alpha^{\mathrm{HF}})$ and $Q^{\mathrm{LF}}_{\mathrm{NN}}(\cdot;\alpha^{\mathrm{LF}})$ minimizing the loss functions

$$\mathcal{L}^{\mathrm{HF}}_{\mathrm{NN}}(\alpha^{\mathrm{HF}}) = \frac{1}{N}\sum_{n=1}^{N}\left|Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}_n) - Q^{\mathrm{HF}}_{\mathrm{NN}}(\boldsymbol{\xi}^{\mathrm{HF}}_n;\alpha^{\mathrm{HF}})\right|,$$
$$\mathcal{L}^{\mathrm{LF}}_{\mathrm{NN}}(\alpha^{\mathrm{LF}}) = \frac{1}{N}\sum_{n=1}^{N}\left|Q^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}_n) - Q^{\mathrm{LF}}_{\mathrm{NN}}(\boldsymbol{\xi}^{\mathrm{LF}}_n;\alpha^{\mathrm{LF}})\right|.$$

We notice that these approximations might not be accurate if the number N of data points is not sufficiently large. On the other hand, if the surrogate models were suitably accurate, then it would make more sense to consider the neural network surrogate rather than the low fidelity model. We emphasize here that these surrogate models are employed with the only purpose of helping to find low dimensional subspaces where the models vary the most, and should not be used to approximate the models themselves for the evaluations needed to compute the multifidelity estimator. Hence, even if the approximation provided by the neural networks surrogates can be quite poor, and this is a problem for identifying the best lower-dimensional subspaces, it can still be sufficient for capturing lower-dimensional manifolds that increase the correlation between the models.

3.2 Coupling MFMC with autoencoders and normalizing flows

A strong limitation of the method presented in the previous section is that it relies on linear dimensionality reduction. In this section, we extend the previous methodology by replacing active subspaces with the supervised autoencoders introduced in Section 2.3, and therefore admitting nonlinear transformations for dimensionality reduction. Although nonlinear transformations are more expressive, they do not preserve Gaussianity, and consequently the normalizing flows used to map the input distributions into standard Gaussian become redundant. Hence, given samples $\{\boldsymbol{\xi}^{\mathrm{HF}}_n\}_{n=1}^N \sim \mu^{\mathrm{HF}}$ and $\{\boldsymbol{\xi}^{\mathrm{LF}}_n\}_{n=1}^N \sim \mu^{\mathrm{LF}}$ from the input distributions, we first learn the autoencoders with r-dimensional latent variables

$$\begin{cases}\mathcal{E}^{\mathrm{HF}}(\cdot;\phi^{\mathrm{HF}}_E)\colon \mathbb{R}^{d^{\mathrm{HF}}} \to \mathbb{R}^r \\ \mathcal{D}^{\mathrm{HF}}(\cdot;\phi^{\mathrm{HF}}_D)\colon \mathbb{R}^{r} \to \mathbb{R}^{d^{\mathrm{HF}}}\end{cases} \quad \text{and} \quad \begin{cases}\mathcal{E}^{\mathrm{LF}}(\cdot;\phi^{\mathrm{LF}}_E)\colon \mathbb{R}^{d^{\mathrm{LF}}} \to \mathbb{R}^r \\ \mathcal{D}^{\mathrm{LF}}(\cdot;\phi^{\mathrm{LF}}_D)\colon \mathbb{R}^{r} \to \mathbb{R}^{d^{\mathrm{LF}}}\end{cases}, \tag{3.6}$$

by minimizing the loss functions

$$\mathcal{L}^{\mathrm{HF}}_{\mathrm{AE}}(\phi^{\mathrm{HF}}_E,\phi^{\mathrm{HF}}_D) = \frac{1}{N}\sum_{n=1}^{N}\left|Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}_n) - Q^{\mathrm{HF}}(\mathcal{D}^{\mathrm{HF}}(\mathcal{E}^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}_n;\phi^{\mathrm{HF}}_E);\phi^{\mathrm{HF}}_D))\right|,$$
$$\mathcal{L}^{\mathrm{LF}}_{\mathrm{AE}}(\phi^{\mathrm{LF}}_E,\phi^{\mathrm{LF}}_D) = \frac{1}{N}\sum_{n=1}^{N}\left|Q^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}_n) - Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}(\mathcal{E}^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}_n;\phi^{\mathrm{LF}}_E);\phi^{\mathrm{LF}}_D))\right|.$$

The latent variables of the autoencoders

$$\boldsymbol{x}^{\mathrm{HF}}_n = \mathcal{E}^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}_n;\phi^{\mathrm{HF}}_E) \in \mathbb{R}^r \qquad \text{and} \qquad \boldsymbol{x}^{\mathrm{LF}}_n = \mathcal{E}^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}_n;\phi^{\mathrm{LF}}_E) \in \mathbb{R}^r$$

now share the same dimensionality, but not the same probability distribution. Therefore, in order to create a shared space, we construct two normalizing flows which map the

distributions of the latent spaces of the two autoencoders into a standard Gaussian. In particular, let $\nu^{\mathrm{HF}} = \mathcal{E}^{\mathrm{HF}}(\cdot;\phi_E^{\mathrm{HF}})_{\#}\mu^{\mathrm{HF}}$ and $\nu^{\mathrm{LF}} = \mathcal{E}^{\mathrm{LF}}(\cdot;\phi_E^{\mathrm{LF}})_{\#}\mu^{\mathrm{LF}}$, and consider the normalizing flows $\mathcal{S}^{\mathrm{HF}}(\cdot;\sigma^{\mathrm{HF}})\colon \mathbb{R}^r \to \mathbb{R}^r$ and $\mathcal{S}^{\mathrm{LF}}(\cdot;\sigma^{\mathrm{LF}})\colon \mathbb{R}^r \to \mathbb{R}^r$ such that

$$\mathcal{S}^{\mathrm{HF}}(\cdot;\sigma^{\mathrm{HF}})_{\#}\nu^{\mathrm{HF}} = \mathcal{N}(\mathbf{0}, I_r) \qquad \text{and} \qquad \mathcal{S}^{\mathrm{LF}}(\cdot;\sigma^{\mathrm{LF}})_{\#}\nu^{\mathrm{LF}} = \mathcal{N}(\mathbf{0}, I_r), \tag{3.7}$$

and which are obtained minimizing the loss functions

$$\mathcal{L}_{\mathrm{NF}}^{\mathrm{HF}}(\sigma^{\mathrm{HF}}) = \sum_{n=1}^{N}\left[\frac{1}{2}\left\|\mathcal{S}^{\mathrm{HF}}(\boldsymbol{x}_n^{\mathrm{HF}};\sigma^{\mathrm{HF}})\right\|^2 - \log\left|\det\nabla\mathcal{S}^{\mathrm{HF}}(\boldsymbol{x}_n^{\mathrm{HF}};\sigma^{\mathrm{HF}})\right|\right],$$
$$\mathcal{L}_{\mathrm{NF}}^{\mathrm{LF}}(\sigma^{\mathrm{LF}}) = \sum_{n=1}^{N}\left[\frac{1}{2}\left\|\mathcal{S}^{\mathrm{LF}}(\boldsymbol{x}_n^{\mathrm{LF}};\sigma^{\mathrm{LF}})\right\|^2 - \log\left|\det\nabla\mathcal{S}^{\mathrm{LF}}(\boldsymbol{x}_n^{\mathrm{LF}};\sigma^{\mathrm{LF}})\right|\right].$$

Remark 3.2. The autoencoders and the normalizing flows are trained sequentially, meaning that we first compute the best autoencoders and then train the normalizing flows on the resulting latent spaces, in order to get latent variables distributed as standard Gaussians. We would also like to point out that we could have used variational autoencoders which would certainly result in a more regular latent space, but without any guarantees on the distribution of the latent space to be a standard Gaussian, unlike normalizing flow where this is guaranteed by construction. Moreover, we remark that, in principle, the normalizing flows could be replaced by any other map between the latent spaces of the autoencoders of the high-fidelity and low-fidelity models, and we leave the determination of the most efficient method for this task to subsequent work.

We therefore obtain a shared space by applying first the encoder and then the normalizing flow as follows

$$\boldsymbol{z}_n^{\mathrm{HF}} = \mathcal{S}^{\mathrm{HF}}(\boldsymbol{x}_n^{\mathrm{HF}};\sigma^{\mathrm{HF}}) = \mathcal{S}^{\mathrm{HF}}(\mathcal{E}^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}};\phi_E^{\mathrm{HF}});\sigma^{\mathrm{HF}}) \in \mathbb{R}^r,$$
$$\boldsymbol{z}_n^{\mathrm{LF}} = \mathcal{S}^{\mathrm{LF}}(\boldsymbol{x}_n^{\mathrm{LF}};\sigma^{\mathrm{LF}}) = \mathcal{S}^{\mathrm{LF}}(\mathcal{E}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{LF}};\phi_E^{\mathrm{LF}});\sigma^{\mathrm{LF}}) \in \mathbb{R}^r,$$

where both $\boldsymbol{z}_n^{\mathrm{HF}}$ and $\boldsymbol{z}_n^{\mathrm{LF}}$ are distributed accordingly to a standard Gaussian $\mathcal{N}(\mathbf{0}, I_r)$. We recall here that $\boldsymbol{\xi}$ stands for the original input, $\boldsymbol{x}$ is the latent variable of the autoencoder, and $\boldsymbol{z}$ is the corresponding normally distributed variable in the shared space. We remark that the autoencoders do not provide a ranking of the variables in the latent dimension according to their variance contributions like the active subspace technique. Hence, if the latent space is multidimensional, we propose to select the best ordering of the components of the latent variable in terms of the estimated correlation. The next steps are analogous to the previous methodology introduced in Section 3.1. In order to increase the correlation between the high-fidelity and low-fidelity models, and without modifying the former, we construct a new low-fidelity model in which we map the input parameters of the high-fidelity model into the input parameters of the original low-fidelity model by means of the shared space. In particular, we define

$$\mathcal{Q}_{\mathrm{AE}}^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{HF}}) = Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}((\mathcal{S}^{\mathrm{LF}})^{-1}(\mathcal{S}^{\mathrm{HF}}(\mathcal{E}^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}};\phi_E^{\mathrm{HF}});\sigma^{\mathrm{HF}});\sigma^{\mathrm{LF}});\phi_D^{\mathrm{LF}})), \tag{3.8}$$

which yields another new multifidelity estimator

$$\widehat{Q}_{N^{\mathrm{HF}},N^{\mathrm{LF}}}^{\mathrm{MFMC,AE}} = \frac{1}{N^{\mathrm{HF}}}\sum_{n=1}^{N^{\mathrm{HF}}} Q^{\mathrm{HF}}(\boldsymbol{\xi}_n^{\mathrm{HF}}) - \beta\left(\frac{1}{N^{\mathrm{HF}}}\sum_{n=1}^{N^{\mathrm{HF}}}\mathcal{Q}_{\mathrm{AE}}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{HF}}) - \frac{1}{N^{\mathrm{LF}}}\sum_{n=1}^{N^{\mathrm{LF}}}\mathcal{Q}_{\mathrm{AE}}^{\mathrm{LF}}(\boldsymbol{\xi}_n^{\mathrm{HF}})\right), \tag{3.9}$$

which is unbiased since we do not modify the high-fidelity model. In Algorithm 2 we summarize the main steps needed to construct the estimator. The same observations highlighted for the methodology based on active subspaces still hold here. In particular, we notice that, in case we cannot generate new high-fidelity simulations, we can reuse the evaluations employed for training the autoencoder, and change only the low-fidelity simulations. Moreover, without reaching the optimal allocation for N^{HF} and N^{LF} we would not get the optimal variance reduction stated in Remark 2.1, but we would still decrease the variance of the multifidelity estimator due to the higher correlation between Q^{HF} and $\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}$ with respect to Q^{HF} and Q^{LF}.

Algorithm 2: MFMC AE

Input: High fidelity and low fidelity models Q^{HF} and Q^{LF}
Distributions μ^{HF} and μ^{LF} for the input parameters
or samples $\{\boldsymbol{\xi}^{\mathrm{HF}}_n\}_{n=1}^N \sim \mu^{\mathrm{HF}}$ and $\{\boldsymbol{\xi}^{\mathrm{LF}}_n\}_{n=1}^N \sim \mu^{\mathrm{LF}}$ from them
Computational budget $\mathcal{B}$.

Output: Estimation $\widehat{Q}^{\mathrm{MFMC,AE}}_{N^{\mathrm{HF}},N^{\mathrm{LF}}}$ of $\mathbb{E}[Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}})]$.

1: Compute the autoencoders $(\mathcal{E}^{\mathrm{HF}}(\cdot;\phi^{\mathrm{HF}}_E), \mathcal{D}^{\mathrm{HF}}(\cdot;\phi^{\mathrm{HF}}_D))$ and $(\mathcal{E}^{\mathrm{LF}}(\cdot;\phi^{\mathrm{LF}}_E), \mathcal{D}^{\mathrm{LF}}(\cdot;\phi^{\mathrm{LF}}_D))$ in (3.6).

2: Compute the normalizing flows $\mathcal{S}^{\mathrm{HF}}(\cdot;\sigma^{\mathrm{HF}})$ and $\mathcal{S}^{\mathrm{LF}}(\cdot;\sigma^{\mathrm{LF}})$ which satisfy (3.7).

3: Select the best ordering of the components of the latent variable.

4: Define the new low-fidelity model $\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}$ in (3.8).

5: Compute the optimal allocation $N^{\mathrm{HF}}, N^{\mathrm{LF}}$ from a pilot sample.

6: Compute the estimator $\widehat{Q}^{\mathrm{MFMC,AE}}_{N^{\mathrm{HF}},N^{\mathrm{LF}}}$ in (3.9).

Remark 3.3. In order to train the autoencoder, it is necessary to evaluate the models Q^{HF} and Q^{LF} multiple times, and this not only is impossible for the expensive high-fidelity model, but can also be impractical for the cheaper low-fidelity model. Therefore, we propose to train surrogate models $Q^{\mathrm{HF}}_{\mathrm{NN}}$ and $Q^{\mathrm{LF}}_{\mathrm{NN}}$ for Q^{HF} and Q^{LF} based on fully connected neural networks with ReLU activation functions, analogously to what we did in Remark 3.1. We recall that these surrogate models might not be highly accurate, in particular for small amount of data, and therefore should not be used for model approximation, but only with the aim of finding nonlinear subspaces of reduced dimension during the training of the autoencoders. We also recall that even if these surrogate models are not sufficiently accurate to determine the best lower-dimensional manifolds, they can still capture nonlinear subspaces that increase the correlation.

3.2.1 A particular choice for the autoencoder

In this section we restrict ourselves to the particular case where the encoder is the model itself, i.e., $\mathcal{E}^{\mathrm{HF}} = Q^{\mathrm{HF}}$ and $\mathcal{E}^{\mathrm{LF}} = Q^{\mathrm{LF}}$. Then, the decoders $\mathcal{D}^{\mathrm{HF}}$ and $\mathcal{D}^{\mathrm{LF}}$ need to satisfy

$$Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}) \simeq Q^{\mathrm{HF}}(\mathcal{D}^{\mathrm{HF}}(Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}))) \qquad \text{and} \qquad Q^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}) \simeq Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}(Q^{\mathrm{LF}}(\boldsymbol{\xi}^{\mathrm{LF}}))),$$

which imply

$$\boldsymbol{x}^{\mathrm{HF}} \simeq Q^{\mathrm{HF}}(\mathcal{D}^{\mathrm{HF}}(\boldsymbol{x}^{\mathrm{HF}})) \qquad \text{and} \qquad \boldsymbol{x}^{\mathrm{LF}} \simeq Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}(\boldsymbol{x}^{\mathrm{LF}})), \tag{3.10}$$

for all $\boldsymbol{x}^{\mathrm{HF}}$ in the image of Q^{HF} and $\boldsymbol{x}^{\mathrm{LF}}$ in the image of Q^{LF}. It is possible to show that the equalities in (3.10) can be satisfied exactly by choosing $\mathcal{D}^{\mathrm{HF}}$ and $\mathcal{D}^{\mathrm{LF}}$ to be the right inverses of the functions Q^{HF} and Q^{LF}, respectively. Indeed, the right inverse of a function exists if the function is surjective, but we can make the function surjective by restricting its codomain to its image. Moreover, an advantage of this formulation is that it is not required to compute the decoders explicitly because we only need the compositions $Q^{\mathrm{HF}} \circ \mathcal{D}^{\mathrm{HF}}$ and $Q^{\mathrm{LF}} \circ \mathcal{D}^{\mathrm{LF}}$, which correspond to the identity function by equations (3.10), and therefore we do not need to train the autoencoders. In this case, the modified low-fidelity model simplifies to

$$\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}(\boldsymbol{\xi}^{\mathrm{HF}}) = (\mathcal{S}^{\mathrm{LF}})^{-1}(\mathcal{S}^{\mathrm{HF}}(Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}}); \sigma^{\mathrm{HF}}); \sigma^{\mathrm{LF}}).$$

The only limitation of this approach is that the new low-fidelity model $\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}$ depends on the high-fidelity model Q^{HF}, which must therefore be replaced by a cheaper surrogate model based on fully connected neural networks, as already done for the other methods and described in Remarks 3.1 and 3.3. On the other hand, when the quantity of interest is scalar, the fact that the encoder is the model itself implies that the shared space is one-dimensional, which in turn yields that the normalizing flows $\mathcal{S}^{\mathrm{HF}}(\cdot; \sigma^{\mathrm{HF}})$ and $\mathcal{S}^{\mathrm{LF}}(\cdot; \sigma^{\mathrm{LF}})$ are one-dimensional mappings.

3.3 Computational complexity analysis

In this section we analyze the computational budget $\mathfrak{C}$ that we need to spend in order to perform our pipelines. It is clear that this additional computational cost is not required for the standard multifidelity Monte Carlo estimator. Hence, one can argue that this budget might be better invested in high-fidelity samples rather than in methodologies for increasing the correlation between the models. In the next proposition we study when it is worth spending part of the computational budget in building our modified estimators. The result is dependent on the ratio $\eta = \mathfrak{C}/\mathcal{B}$ between the cost for training the networks in the pipelines and the total computational budget, and on the improvement in term of correlation between the models.

Proposition 3.4. *Let ρ be the initial correlation between the high-fidelity and low-fidelity models, and let ρ_{AS} and ρ_{AE} be new correlations after performing the pipelines based on active subspace and autoencoder, respectively. Assume that*

$$|\rho_{\mathrm{A}*}| > |\rho| > \frac{4w}{(1+w)^2},$$

and

$$\frac{\mathfrak{C}}{\mathcal{B}} = \eta < 1 - \frac{\sqrt{1-\rho^2_{\mathrm{A}*}} + \sqrt{w\rho^2_{\mathrm{A}*}}}{\sqrt{1-\rho^2} + \sqrt{w\rho^2}},$$

where $\mathfrak{C}$ is the cost for training the networks, $\mathcal{B}$ is the total computational budget, $\mathrm{A}*$ stands for both AS *and* AE*, and $w = \mathcal{C}^{\mathrm{LF}}/\mathcal{C}^{\mathrm{HF}}$ is the cost ratio between the two fidelities. Then, it holds*

$$\mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{MFMC,A}*}_{\mathcal{B}}\right] < \mathbb{V}\mathrm{ar}\left[\widehat{Q}^{\mathrm{MFMC}}_{\mathcal{B}}\right],$$

where both the estimators are computed assuming a total computational budget $\mathcal{B}$ and solving the optimal allocation problem.

Proof. Using equation (2.3) we have

$$\mathbb{V}\mathrm{ar}\left[\widehat{Q}_{\mathcal{B}}^{\mathrm{MFMC}}\right] = \mathbb{V}\mathrm{ar}\left[\widehat{Q}_{\mathcal{B}}^{\mathrm{MC}}\right]\left(\sqrt{1-\rho^2}+\sqrt{w\rho^2}\right) = \frac{\mathbb{V}\mathrm{ar}\left[Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}})\right]}{\mathcal{B}}\left(\sqrt{1-\rho^2}+\sqrt{w\rho^2}\right). \tag{3.11}$$

Moreover, since the computational budget $\mathfrak{C} = \eta\mathcal{B}$ is spent for building the modified estimators, we then employ equation (2.3) with budget $\mathcal{B} - \mathfrak{C}$, and obtain

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\widehat{Q}_{\mathcal{B}}^{\mathrm{MFMC,A}*}\right] &= \mathbb{V}\mathrm{ar}\left[\widehat{Q}_{\mathcal{B}-\mathfrak{C}}^{\mathrm{MC}}\right]\left(\sqrt{1-\rho_{\mathrm{A}*}^2}+\sqrt{w\rho_{\mathrm{A}*}^2}\right)\\ &= \frac{\mathbb{V}\mathrm{ar}\left[Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}})\right]}{\mathcal{B}-\mathfrak{C}}\left(\sqrt{1-\rho_{\mathrm{A}*}^2}+\sqrt{w\rho_{\mathrm{A}*}^2}\right)\\ &= \frac{\mathbb{V}\mathrm{ar}\left[Q^{\mathrm{HF}}(\boldsymbol{\xi}^{\mathrm{HF}})\right]}{\mathcal{B}(1-\eta)}\left(\sqrt{1-\rho_{\mathrm{A}*}^2}+\sqrt{w\rho_{\mathrm{A}*}^2}\right).\end{aligned} \tag{3.12}$$

Finally, combining equations (3.11) and (3.12) gives the desired result. □

The previous result shows the condition under which we get a benefit in using our approaches. In particular, as long as the correlation increases significantly and the cost for building the modified estimators is sufficiently small compared to the total computational budget, it is better to apply our methodologies rather than using standard multifidelity Monte Carlo. This is true especially for all computationally expensive models that appear in concrete applications, such as the cardiovascular simulations in Section 4.3.

3.4 A theoretical example

The goal of this section is to give a better understanding of the methodologies introduced in the previous sections, by considering a simple example where every computation can be performed analytically. Let the high-fidelity and low-fidelity models $Q^{\mathrm{HF}}, Q^{\mathrm{LF}} \colon \mathbb{R}^2 \to \mathbb{R}$ be the functions

$$Q^{\mathrm{HF}}(x,y) = x + y \qquad \text{and} \qquad Q^{\mathrm{LF}}(x,y) = \frac{x}{2} + 2y,$$

and let the distributions of the input values be $\mu^{\mathrm{HF}} = \mu^{\mathrm{LF}} = \mu = \mathcal{U}([-1,1]^2)$. First, notice that the two models have zero mean and consequently their Pearson correlation coefficient is

$$\rho = \frac{\mathbb{E}\left[(X+Y)\left(\frac{X}{2}+2Y\right)\right]}{\sqrt{\mathbb{E}[(X+Y)^2]\,\mathbb{E}\left[\left(\frac{X}{2}+2Y\right)^2\right]}} = \frac{5}{\sqrt{34}} \simeq 0.86.$$

We now apply our approaches to modify the low-fidelity model and thus increase the correlation coefficient. Let us first consider the method presented in Section 3.1. The normalizing flow, which transforms the uniform distribution μ into a standard two-dimensional Gaussian $\mathcal{N}(0, I_2)$, and its inverse are given by

$$\mathcal{T}(x,y) = \begin{pmatrix}\sqrt{2}\,\mathrm{erf}^{-1}(x)\\ \sqrt{2}\,\mathrm{erf}^{-1}(y)\end{pmatrix} \qquad \text{and} \qquad \mathcal{T}^{-1}(x,y) = \begin{pmatrix}\mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\\ \mathrm{erf}\left(\frac{y}{\sqrt{2}}\right)\end{pmatrix},$$

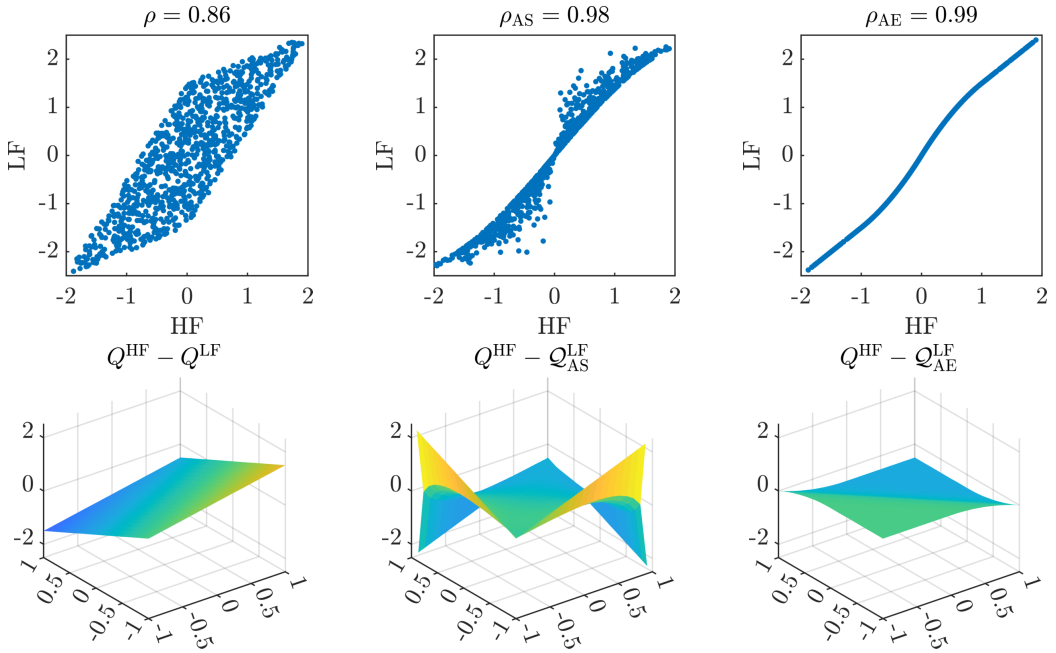

Figure 2: Correlation (top) and difference (bottom) between HF model and original LF model (left), LF models given by the methods based on active subspaces (center) and autoencoders (right).

which is equal for both the high-fidelity and low-fidelity models. We then obtain the modified models $\widetilde{Q}^{\mathrm{HF}}, \widetilde{Q}^{\mathrm{LF}} \colon \mathbb{R}^2 \to \mathbb{R}$

$$\widetilde{Q}^{\mathrm{HF}}(x,y) = Q^{\mathrm{HF}}(\mathcal{T}^{-1}(x,y)) = \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + \operatorname{erf}\left(\frac{y}{\sqrt{2}}\right),$$
$$\widetilde{Q}^{\mathrm{LF}}(x,y) = Q^{\mathrm{LF}}(\mathcal{T}^{-1}(x,y)) = \frac{1}{2}\operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) + 2\operatorname{erf}\left(\frac{y}{\sqrt{2}}\right),$$

whose input distribution is the standard Gaussian, and which give the matrices

$$C^{\mathrm{HF}} = \mathbb{E}\left[\nabla\widetilde{Q}^{\mathrm{HF}}(X,Y)\nabla\widetilde{Q}^{\mathrm{HF}}(X,Y)^{\top}\right] = \frac{2}{\pi}\,\mathbb{E}\left[\begin{pmatrix} e^{-X^2} & e^{-\frac{X^2+Y^2}{2}} \\ e^{-\frac{X^2+Y^2}{2}} & e^{-Y^2} \end{pmatrix}\right] = \begin{pmatrix} \frac{2}{\pi\sqrt{3}} & \frac{1}{\pi} \\ \frac{1}{\pi} & \frac{2}{\pi\sqrt{3}} \end{pmatrix},$$

and

$$C^{\mathrm{LF}} = \mathbb{E}\left[\nabla\widetilde{Q}^{\mathrm{LF}}(X,Y)\nabla\widetilde{Q}^{\mathrm{LF}}(X,Y)^{\top}\right] = \frac{2}{\pi}\,\mathbb{E}\left[\begin{pmatrix} \frac{1}{4}e^{-X^2} & e^{-\frac{X^2+Y^2}{2}} \\ e^{-\frac{X^2+Y^2}{2}} & 4e^{-Y^2} \end{pmatrix}\right] = \begin{pmatrix} \frac{1}{2\pi\sqrt{3}} & \frac{1}{\pi} \\ \frac{1}{\pi} & \frac{8}{\pi\sqrt{3}} \end{pmatrix}.$$

The corresponding one-dimensional active subspaces are

$$W_A^{\mathrm{HF}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \text{and} \qquad W_A^{\mathrm{LF}} = \frac{1}{\sqrt{273 + 15\sqrt{273}}}\begin{pmatrix} 2\sqrt{6} \\ \frac{15+\sqrt{273}}{\sqrt{2}} \end{pmatrix},$$

and are used in the definition of the new low-fidelity model

$$\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AS}}(x,y) = Q^{\mathrm{LF}}(\mathcal{T}^{-1}(W_A^{\mathrm{LF}}(W_A^{\mathrm{HF}})^{\top}\mathcal{T}(x,y))),$$

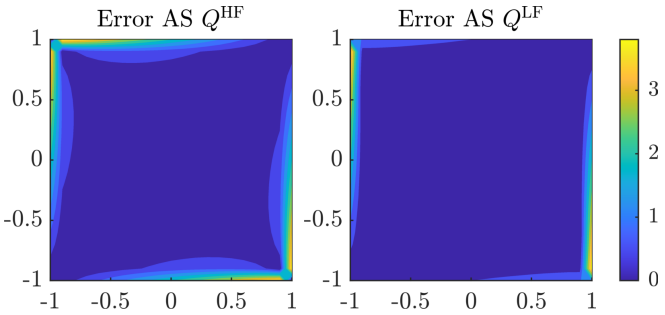

Figure 3: Truncation error introduced reducing the dimensionality of the models through active subspaces and normalizing flow, for the theoretical example. The error is computed as $\left|Q^{*\mathrm{F}}(x,y) - Q^{*\mathrm{F}}(\mathcal{T}^{-1}(W_A^{*\mathrm{F}}(W_A^{*\mathrm{F}})^\top \mathcal{T}(x,y)))\right|$, where $*\mathrm{F}$ stands for both HF and LF.

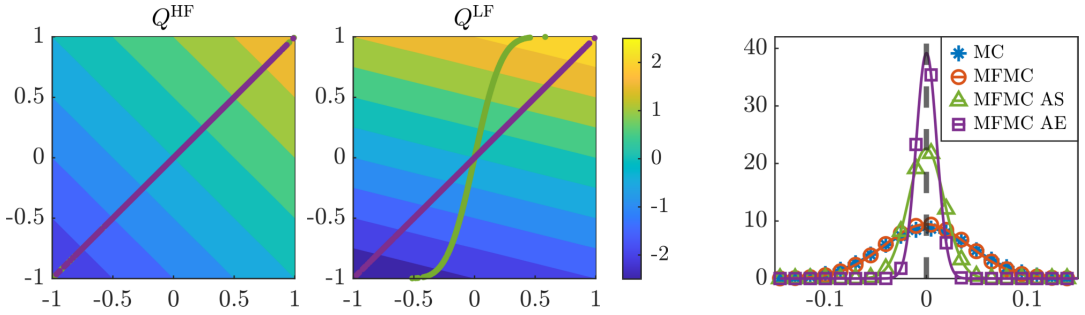

Figure 4: Left: contour plot of Q^{HF} and Q^{LF} with the corresponding dominant directions determined by the methods based on active subspaces (in green) and autoencoders (in purple), for the theoretical example. Right: comparison between our methods (MFMC AS and MFMC AE) with standard (multifidelity) Monte Carlo (MC and MFMC), for the theoretical example.

which yields an improved correlation

$$\rho_{\mathrm{AS}} = \frac{\mathbb{C}\mathrm{ov}(Q^{\mathrm{HF}}(X,Y), \mathcal{Q}^{\mathrm{LF}}_{\mathrm{AS}}(X,Y))}{\sqrt{\mathbb{V}\mathrm{ar}[Q^{\mathrm{HF}}(X,Y)]\,\mathbb{V}\mathrm{ar}[\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AS}}(X,Y)]}} \simeq 0.98 > \rho.$$

We remark that the correlation coefficient is computed numerically employing 10^8 samples from the distribution μ. Let us now focus on the method presented in Section 3.2. Consider the following choice for the autoencoders

$$\begin{cases} \mathcal{E}^{\mathrm{HF}} : [-1,1]^2 \to [-2,2], & \mathcal{E}^{\mathrm{HF}}(x,y) = x + y, \\ \mathcal{D}^{\mathrm{HF}} : [-2,2] \to [-1,1]^2, & \mathcal{D}^{\mathrm{HF}}(z) = \begin{pmatrix} \frac{z}{2} \\ \frac{z}{2} \end{pmatrix}, \end{cases}$$

$$\begin{cases} \mathcal{E}^{\mathrm{LF}} : [-1,1]^2 \to \left[-\frac{5}{2}, \frac{5}{2}\right], & \mathcal{E}^{\mathrm{LF}}(x,y) = \frac{x}{2} + 2y, \\ \mathcal{D}^{\mathrm{LF}} : \left[-\frac{5}{2}, \frac{5}{2}\right] \to [-1,1]^2, & \mathcal{D}^{\mathrm{LF}}(z) = \begin{pmatrix} \frac{2}{5}z \\ \frac{2}{5}z \end{pmatrix}, \end{cases}$$

and notice that the original models can be reconstructed exactly, i.e.,

$$Q^{\mathrm{HF}}(x,y) = Q^{\mathrm{HF}}(\mathcal{D}^{\mathrm{HF}}(\mathcal{E}^{\mathrm{HF}}(x,y))) \qquad \text{and} \qquad Q^{\mathrm{LF}}(x,y) = Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}(\mathcal{E}^{\mathrm{LF}}(x,y))).$$

We remark that the autoencoders which give an exact representation of the functions are not unique, for example we could rescale the encoders by a constant $c > 0$ and compute the decoders accordingly, without affecting the final correlation. We now have to find a normalizing flow from the latent space of each model to a standard one-dimensional Gaussian $\mathcal{N}(0,1)$. From the encoders $\mathcal{E}^{\mathrm{HF}}$ and $\mathcal{E}^{\mathrm{LF}}$, we deduce that the distributions of the latent spaces of the high-fidelity and low-fidelity models are

$$\nu^{\mathrm{HF}} = \mathcal{T}ri(-2,0,+2) \qquad \text{and} \qquad \nu^{\mathrm{LF}} = \mathcal{T}rap\left(-\frac{5}{2},-\frac{3}{2},+\frac{3}{2},+\frac{5}{2}\right),$$

where $\mathcal{T}ri$ and $\mathcal{T}rap$ stand for triangular and trapezoidal distribution, respectively. Hence, following [16], the normalizing flows are given by

$$\mathcal{S}^{\mathrm{HF}}(z) = \sqrt{2}\,\mathrm{erf}^{-1}(U^{\mathrm{HF}}(z)) \qquad \text{and} \qquad \mathcal{S}^{\mathrm{LF}}(z) = \sqrt{2}\,\mathrm{erf}^{-1}(U^{\mathrm{LF}}(z)),$$

where

$$U^{\mathrm{HF}}(z) = \begin{cases} \frac{1}{4}(2+z)^2 - 1, & -2 \le z \le 0, \\ 1 - \frac{1}{4}(2-z)^2, & 0 \le z \le 2, \end{cases}$$

$$U^{\mathrm{LF}}(z) = \begin{cases} \frac{1}{4}\left(\frac{5}{2}+z\right)^2 - 1, & -\frac{5}{2} \le z \le -\frac{3}{2}, \\ \frac{z}{2}, & -\frac{3}{2} \le z \le \frac{3}{2}, \\ 1 - \frac{1}{4}\left(\frac{5}{2}-z\right)^2, & \frac{3}{2} \le z \le \frac{5}{2}, \end{cases}$$

These transformations are used in the definition of the new low-fidelity model

$$\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}(x,y) = Q^{\mathrm{LF}}(\mathcal{D}^{\mathrm{LF}}((\mathcal{S}^{\mathrm{LF}})^{-1}(\mathcal{S}^{\mathrm{HF}}(\mathcal{E}^{\mathrm{HF}}(x,y))))),$$

which yields an improved correlation

$$\rho_{\mathrm{AE}} = \frac{\mathbb{C}\mathrm{ov}(Q^{\mathrm{HF}}(X,Y), \mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}(X,Y))}{\sqrt{\mathbb{V}\mathrm{ar}[Q^{\mathrm{HF}}(X,Y)]\,\mathbb{V}\mathrm{ar}[\mathcal{Q}^{\mathrm{LF}}_{\mathrm{AE}}(X,Y)]}} \simeq 0.99 > \rho_{\mathrm{AS}} > \rho.$$

The correlation coefficient is computed numerically employing 10^8 samples from the distribution μ, in the same way we did for the method based on active subspaces. In Fig. 2 we plot the correlation between the high-fidelity model and both the original and the new low-fidelity models, together with their difference. We notice that employing a nonlinear transformation results in a larger correlation coefficient, i.e., $\rho_{\mathrm{AE}} > \rho_{\mathrm{AS}}$, which cannot be obtained by means of a linear transformation, and a better approximation of the high-fidelity model, since the differnce is closer to zero. These plots also show that active subspaces introduce a small truncation error, which is not produced by the autoencoder, as seen in Fig. 3. Moreover, in Fig. 4 we plot the dominant subspaces obtained by employing our methodologies. We observe that the two approaches find the same subspace for the high-fidelity model, while they provide different subspaces for the low-fidelity one. The latter is therefore responsible for the better correlation between the models. Finally, still in Fig. 4 we compare standard Monte Carlo (MC) and multifidelity Monte Carlo (MFMC) with our two approaches (MFMC AS) and (MFMC AE). We assume a ratio between the costs of the models equal to $\mathcal{C}^{\mathrm{LF}} = 0.01\mathcal{C}^{\mathrm{HF}}$ to mirror cost differences in realistic applications, and we then set a budget of 100 HF simulations and 20000 LF simulations, which is equivalent to the cost of 300 HF simulations. We observe that both our methods outperform standard techniques, and, in particular, the methodology with the autoencoder achieves a smaller variance with respect to the active subspace due to a larger correlation between the high-fidelity and the reduced low-fidelity models.

Remark 3.5. Notice that equation (2.3) still holds true for the estimators in (3.5) and (3.9). Therefore, if we manage to increase the correlation between the high-fidelity and low-fidelity models, then we also improve the variance of the resulting estimators. For the active subspace technique, and in general for linear approaches, it is reasonable to assume that if we align the important directions of different models, then their correlation should be larger along those directions than in the original space. The intuition for this approach is provided in previous literature [12, 45]; specifically, in [45, Proposition 4.4], it is shown how the re-arrangement of the variables leads to an increased correlation in the linear case. Beyond the intuition or the quantitative analysis presented in [45] under simplifying assumptions, we also note this idea has been adopted successfully on non-trivial aerospace applications with high-dimensional inputs, see, e.g., [11, 13]. On the other hand, even if we do not have any theoretical guarantee that nonlinear lower-dimensional manifolds can increase the correlation, we expect them to behave similarly if mapped appropriately. In this work, we consider autoencoders as a nonlinear extension to linear dimensionality reduction, and the following numerical experiments highlight the possibility to improve the performance of multifidelity Monte Carlo estimators whenever a nonlinear manifold provides a more parsimonious representation than a linear subspace for the input-to-output map in at least one of the models. Moreover, we can always verify whether the new correlation, which can be estimated through a pilot sample as demonstrated in the paper, is larger than the original correlation. If this does not hold, then we can employ standard multifidelity Monte Carlo estimators, so that we can guarantee no reduction in performance apart from the negligible cost increase (compared to the models' evaluations) of the dimension reduction and normalizing flows steps. We finally note that one could train the autoencoders for both the high-fidelity and low-fidelity models simultaneously, and including a term in the loss function that maximizes the resulting correlation.

4 Numerical experiments

In this section we demonstrate the advantages of our approaches through a series of test cases. We first consider analytic functions which allow us to explore the properties of the methods, and then focus on a reaction-diffusion equation. Finally, we consider cardiovascular simulations as an example of a computationally expensive model with concrete applications and for which only a small amount of data can be available. We note that the dashed lines representing the "true" mean in the following plots is given by the average of the Monte Carlo estimator.

Remark 4.1. For all the neural networks appearing in the pipelines, after normalizing the input and output values in the interval $[-1, 1]$, we perform a hyperparameter tuning for the number of layers, number of neurons per layer, learning rate, and exponential scheduler step. In particular, we tune the hyperparamters which appear in the autoencoder, the surrogate models, and the normalizing flows. The hyperparameters are optimized sequentially employing the Optuna optimization framework [1] monitoring the validation loss (20% of the dataset). In particular, we initially find the best parameters for the surrogate models, if necessary, and then use these parameters in the training process for the autoencoder. Then, once we have selected the best parameters for the autoencoder, we employ these values to compute the latent space from which we learn the normalizing flows and their corresponding hyperparameters. In the following numerical experiments, we constrain the number of layers in $\{1, \dots, 4\}$, the number of neurons per layer in $\{1, \dots, 16\}$, the learning rate in $[10^{-4}, 10^{-2}]$, and the exponential scheduler step in $[0.999, 0.9999]$.

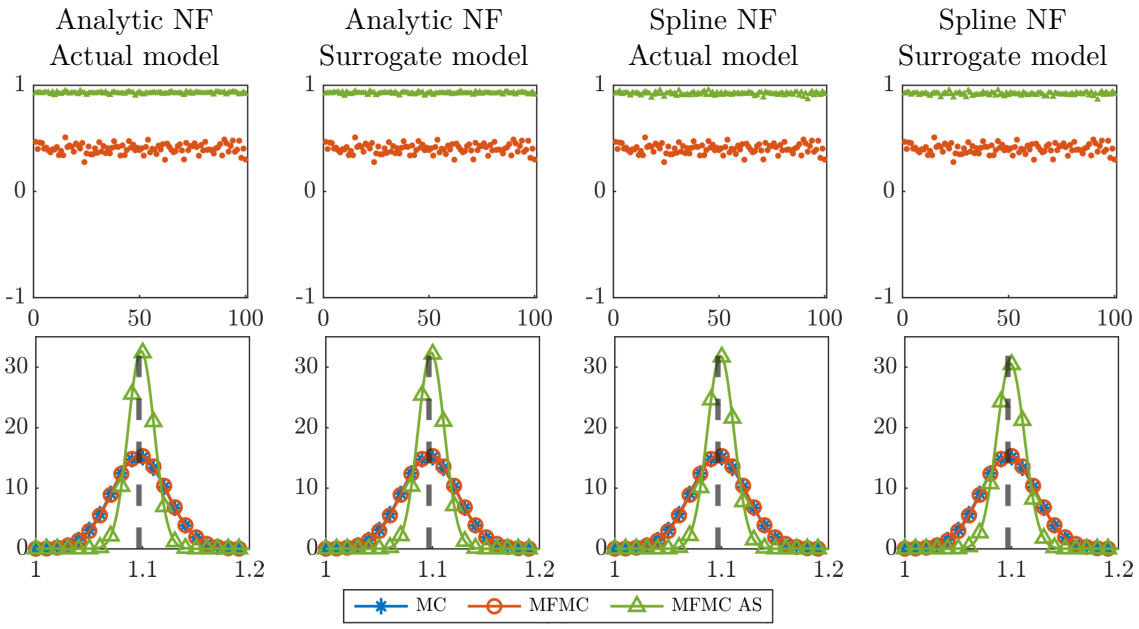

Figure 5: Comparison between our method based on active subspaces (MFMC AS) with standard (multifidelity) Monte Carlo (MC and MFMC), for the case of analytic functions. The normalizing flow can be either exact (Analytic NF) or estimated using splines (Spline NF), and the gradient to compute the active subspace can be either obtained with the analytic gradient (Actual model) or through a surrogate model given by a neural network (Surrogate model). Top: Pearson correlation coefficients for 100 different repetitions. Bottom: approximated distributions of the estimators using 100 samples

Moreover, we train all the neural networks for 5000 epochs with the Adam optimizer [20], and we perform 100 independent repetitions of the entire procedure in order to illustrate its overall variability. The major computational cost for obtaining the networks is given by the hyperparameter tuning, which is however done as a preliminary step, before performing the multifidelity uncertainty propagation pipeline. We also notice that, as highlighted in Section 3.3, the cost for training all the networks in the pipelines is negligible with respect to the cost of high-fidelity and low-fidelity simulations, in particular for computationally expensive models, and therefore we do not include this cost in the comparison between our approaches and standard (multifidelity) Monte Carlo estimators.

4.1 Analytic functions

Inspired by [12], we consider the following functions as high-fidelity and low-fidelity models

$$Q^{\mathrm{HF}}(x,y) = e^{0.7x+0.3y} + 0.15\sin(2\pi x) \quad \text{and} \quad Q^{\mathrm{LF}}(x,y) = e^{0.01x+0.99y} + 0.15\sin(3\pi y),$$

with input distributions $\mu^{\mathrm{HF}} = \mu^{\mathrm{LF}} = \mu = \mathcal{U}([-1,1]^2)$, and we aim to estimate

$$\mathbb{E}[Q^{\mathrm{HF}}(x,y)] = \frac{25}{21}\left(e^{7/10} - e^{-7/10}\right)\left(e^{3/10} - e^{-3/10}\right).$$

We assume a cost of our low-fidelity model equal to $\mathcal{C}^{\mathrm{LF}} = 0.01\mathcal{C}^{\mathrm{HF}}$ ($w = 0.01$) to mirror cost differences in realistic applications. We notice that in this case gradients and normalizing flow needed for the method presented in Section 3.1 can be computed analytically, and are

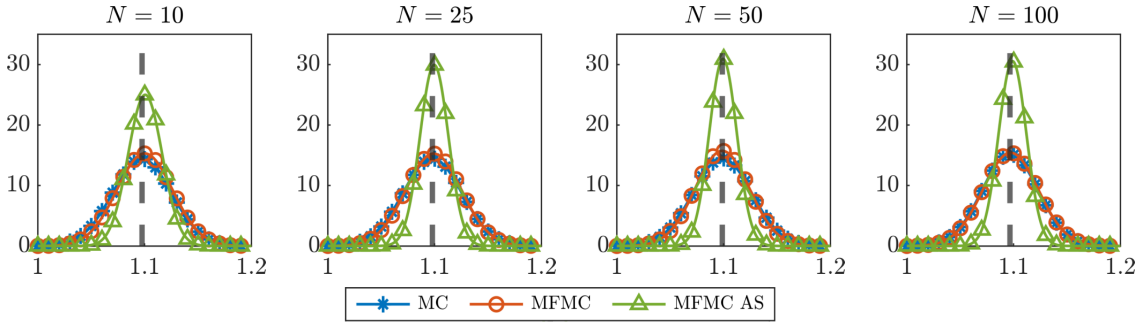

Figure 6: Comparison between our method based on active subspaces (MFMC AS) with standard (multifidelity) Monte Carlo (MC and MFMC), varying the number of data points, for the case of analytic functions. The normalizing flow is estimated using splines, and the gradient to compute the active subspace is obtained through a surrogate model given by a neural network.

given by

$$\begin{aligned} \nabla Q^{\mathrm{HF}}(x,y) &= \begin{pmatrix} 0.7e^{0.7x+0.3y} + 0.3\pi\cos(2\pi x) \\ 0.3\exp^{0.7x+0.3y} \end{pmatrix}, \\ \nabla Q^{\mathrm{LF}}(x,y) &= \begin{pmatrix} 0.01e^{0.01x+0.99y} \\ 0.99\exp^{0.01x+0.99y} + + 0.45\pi\cos(3\pi y) \end{pmatrix}, \end{aligned} \tag{4.1}$$

and

$$\mathcal{T}^{\mathrm{HF}}(x,y) = \mathcal{T}^{\mathrm{LF}}(x,y) = \mathcal{T}(x,y) = \begin{pmatrix} \sqrt{2}\,\mathrm{erf}^{-1}(x) \\ \sqrt{2}\,\mathrm{erf}^{-1}(y) \end{pmatrix}. \tag{4.2}$$

In this section we study the properties of our methodologies and see how they perform on simple examples. We compute the mean value and the standard deviation of the estimator of the quantity of interest employing standard (multifidelity) Monte Carlo estimators and our techniques, and then we plot the approximated Gaussian distributions of the estimators constructed from the approximated mean and variance. In order to take into account all the variance of the methods, we first get a pilot sample from the distributions of the input parameters, which we employ to get the best hyperparameters for the networks as outlined in Remark 4.1, and to train them. Then, we compute the optimal allocation using the pilot sample and setting a computational budget of 300 high-fidelity simulations, and finally we discard the pilot sample and draw new samples from which we obtain the multifidelity estimation with a one-dimensional shared subspace. This procedure is repeated 100 times.

Let us now focus on the method in Section 3.1. In Fig. 5 we consider four different cases, setting the size of the pilot sample equal to 100. In the first column we leverage the fact that we know the exact gradient of the functions (4.1) and the exact normalizing flow from the input distributions (4.2), and we plot the results in the ideal setting where we can employ them. In the second and third columns, we add one level of complexity at a time, by first computing the gradients through surrogate models based on fully connected neural networks, and then by training a spline-based normalizing flow. Finally, in the last column we consider the most general case, where we do not have any a priori knowledge of the gradients and the normalizing flow, which is actually employed in concrete applications. Both from the correlation values and the variances of the output distributions, we observe that in all four cases our approach outperforms standard (multifidelity) Monte Carlo techniques. In particular, we notice that the presence of the surrogate model seems not to affect the final results, and this is due to the fact that the functions representing the high-fidelity

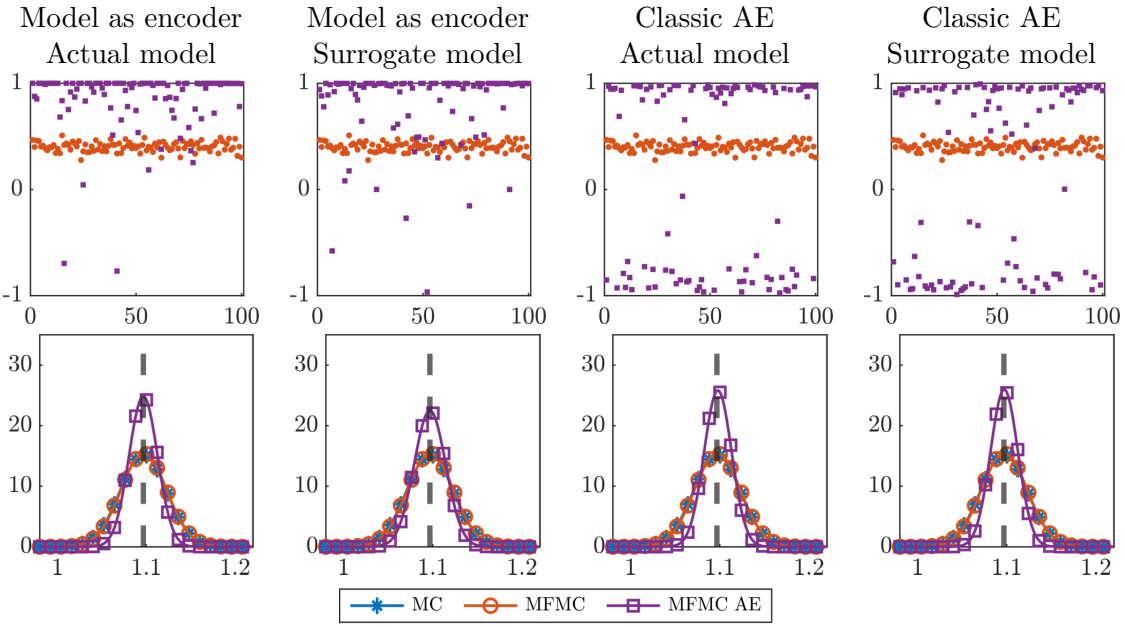

Figure 7: Comparison between our method based on autoencoders (MFMC AE) with standard (multifidelity) Monte Carlo (MC and MFMC), for the case of analytic functions. The encoder can be either fixed (Model as encoder) or computed (Classic AE), and for the training of the autoencoder we can use either the function (Actual model) or a surrogate model given by a neural network (Surrogate model). Top: Pearson correlation coefficients for 100 different repetitions. Bottom: approximated distributions of the estimators using 100 samples.

and low-fidelity models can be easily approximated by fully connected neural networks. We also note a slight increase in the variance when we replace the exact normalizing flow with its spline-based approximations, which, nevertheless, does not deteriorates the final output. Moreover, in Fig. 6 we compare the results varying the size of the pilot sample for the general case where we do not use the analytic gradients or the normalizing flow. We observe that the final variance of the estimator increases only when a significantly small pilot sample is drawn, meaning that in this case the estimator is not strongly sensitive to the number of data, and that it is not necessary to find the exact active subspace to achieve variance reduction, as long as the approximated important direction is not highly different from the real one.

We repeat similar experiments for the method in Section 3.2 with RealNVP as normalizing flow. In Fig. 7 we set the sample size equal to 100, and we consider four different cases. In the first two columns we fix the encoder to be the actual model as described in Section 3.2.1, while in the last two columns we study the general methodology where the autoencoder is trained using the available data. In both cases we use either the model itself or a surrogate model based on fully connected neural networks to compute the encoder or train the autoencoder, respectively. Similarly to the other approach, we observe that our technique is able to improve standard (multifidelity) Monte Carlo estimators, and that the presence of the surrogate model does not seem to affect the final results. Moreover, we do not notice a significant difference between the standard autoencoder approach and the one where the encoder is equal to the model. In Fig. 8 we also compare the results varying the size of the pilot sample for the last case where the autoencoder is trained using the surrogate model. We notice that the variance of the estimator is smaller when the

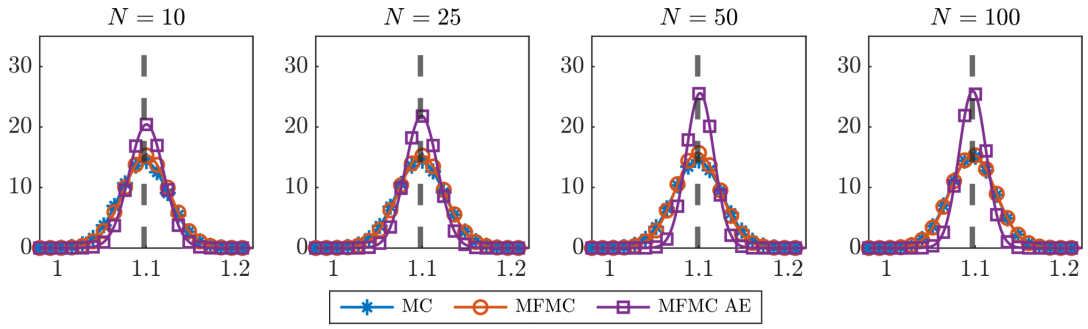

Figure 8: Comparison between our method based on autoencoders (MFMC AE) with standard (multifidelity) Monte Carlo (MC and MFMC), varying the number of data points, for the case of analytic functions. The autoencoder is trained using a surrogate model given by a neural network

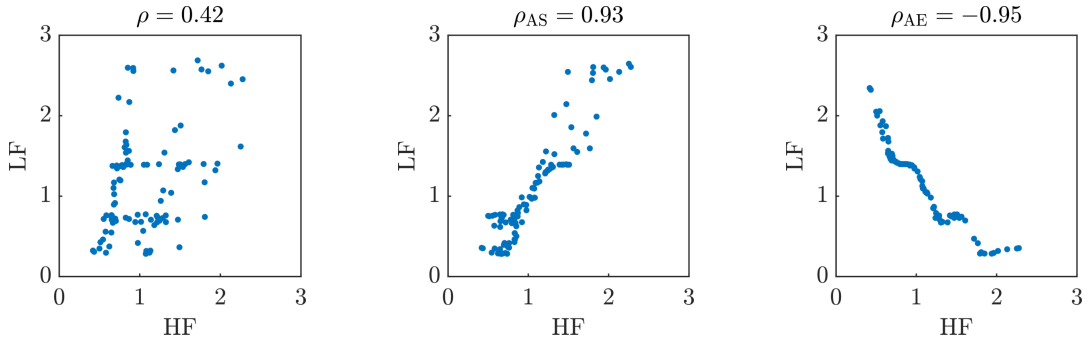

Figure 9: Correlation between HF model and original LF model (left), LF models given by the methods based on active subspaces (center) and autoencoders (right), for the case of analytic functions.

number of data in the pilot sample is larger, i.e., if we have enough information to find the nonlinear subspace where the models vary the most. Finally, in Fig. 9 we show for a particular sample how the correlation increases from the original low-fidelity model to the new reduced low-fidelity models obtained applying our methodologies. It is interesting to notice that the autoencoder seems to introduce a nonnegligible bias, compared to the active subspace. Nevertheless, this is not important for the performance of the algorithm that only depends on the correlation between the models. We finally remark that, in order for the method with the autoencoder to be able to have a better performance with respect to the method with active subspaces, a larger number of data or a more complex problem is necessary, as we will see in the next examples.

4.2 Reaction-diffusion equation

In this section we work with a more complex example taken from the PDEBench repository [41], which has applications in real-world problems, i.e., biological pattern formation [43]. In particular, we consider the two-dimensional reaction-diffusion equation

$$\begin{aligned} \frac{\partial u(t,x,y)}{\partial t} &= D_u \frac{\partial^2 u(t,x,y)}{\partial x^2} + D_u \frac{\partial^2 u(t,x,y)}{\partial y^2} + R_u(u(t,x,y), v(t,x,y)), \\ \frac{\partial v(t,x,y)}{\partial t} &= D_v \frac{\partial^2 v(t,x,y)}{\partial x^2} + D_v \frac{\partial^2 v(t,x,y)}{\partial y^2} + R_v(u(t,x,y), v(t,x,y)), \end{aligned} \tag{4.3}$$

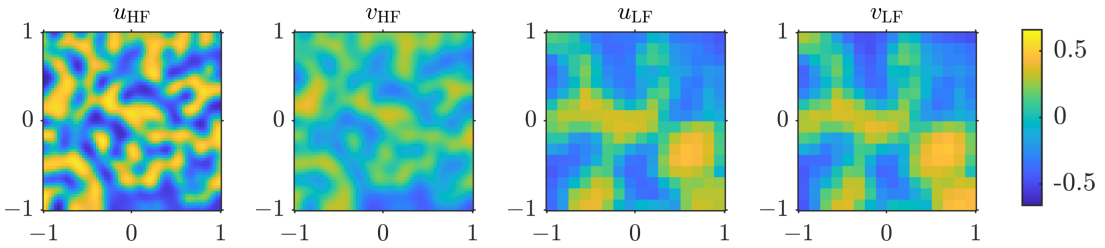

Figure 10: High-fidelity and low-fidelity solutions of the reaction-diffusion equation at the final time $T = 4$.

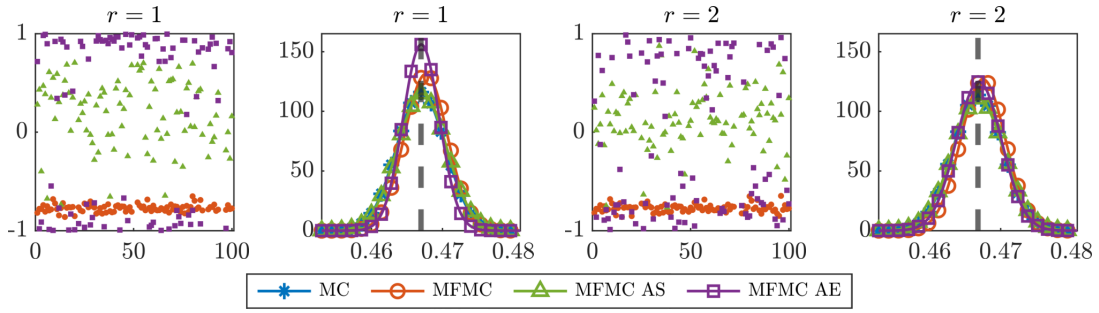

Figure 11: Comparison between our methods (MFMC AS and MFMC AE) with standard (multifidelity) Monte Carlo (MC and MFMC) for the reaction-diffusion equation, varying the reduced dimension $r = 1, 2$.

with no-flow Neumann boundary conditions

$$D_u \frac{\partial u(t,x,y)}{\partial x} = 0, \qquad D_u \frac{\partial u(t,x,y)}{\partial y} = 0, \qquad t \in (0,T], \quad x,y \in (-L,L),$$
$$D_v \frac{\partial v(t,x,y)}{\partial x} = 0, \qquad D_v \frac{\partial v(t,x,y)}{\partial y} = 0, \qquad t \in (0,T], \quad x,y \in (-L,L),$$

where $u, v \colon (0,T] \times (-L,L) \times (-L,L)$ are called the activator and the inhibitor, $T > 0$ is the final time, $L > 0$ is the size of the domain, $D_u, D_v > 0$ are the diffusion coefficients, and R_u, R_v are the reaction functions

$$R_u(u,v) = u - u^3 - k - v,$$
$$R_v(u,v) = u - v,$$

with $k > 0$ and which are equivalent to the well-known Fitzhugh-Nagumo equations [21]. The initial condition is generated as standard normal random noise $u(0,x,y), v(0,x,y) \sim \mathcal{N}(0,1)$ for $x \in (-L,L)$ and $y \in (-L,L)$ only once and then fixed. We consider as our quantity of interest the sum of the averages of the absolute value of the activator and the inhibitor at the final time

$$Q = \frac{1}{4L^2} \int_{-L}^{L} \int_{-L}^{L} |u(T,x,y)| \, \mathrm{d}x \, \mathrm{d}y + \frac{1}{4L^2} \int_{-L}^{L} \int_{-L}^{L} |v(T,x,y)| \, \mathrm{d}x \, \mathrm{d}y,$$

and we set the final time $T = 4$ and the size of the domain $L = 1$. The equations are solved employing the finite volume method for space discretization and the fourth order Runge–Kutta method for time integration. The high-fidelity model is obtained by setting 64 finite volume cells in each direction and running the simulation for 400 time steps, while for the low-fidelity model we use 16 finite volume cells in each direction and run for 100 time steps,

which yields a cost ratio of $w = 0.1$. Moreover, in the low-fidelity model we replace the diffusion coefficients D_u and D_v by their average value, i.e., $\bar{D} = (D_u + D_v)/2$. The input parameters for the high-fidelity model are the diffusion coefficients D_u, D_v and the reaction coefficient k, while for the low-fidelity model we have $\bar{D}$ and k, and therefore we are in the framework where the two models have a dissimilar parameterization. In Fig. 10 we plot the solutions u and v of equations (4.3) for both the high-fidelity and low-fidelity models, setting the parameters $D_u = 10^{-3}, D_v = 5 \cdot 10^{-3}, k = 10^{-3}$, and $\bar{D} = 3 \cdot 10^{-3}$. The input probability distribution for the forward uncertainty propagation study is such that k is independent of D_u and D_v, $k \sim \mathcal{U}([0.5 \cdot 10^{-3}, 1.5 \cdot 10^{-3}])$, and D_u, D_v are uniformly distributed in the triangle with vertices $\{(0.25 \cdot 10^{-3}, 4 \cdot 10^{-3}), (1.75 \cdot 10^{-3}, 5 \cdot 10^{-3}), (1 \cdot 10^{-3}, 6 \cdot 10^{-3})\}$. We remark that this is an example of correlated random inputs, since D_u and D_v in the high-fidelity model are uniformly distributed on a triangular support, leading to correlated parameters. The distribution of $\bar{D}$ is then obtained accordingly. One of the challenges when working with dependent inputs relates to the difficulty of enforcing that the new low-fidelity samples generated by the pipeline are defined over the same support as the original low-fidelity inputs. To overcome this problem, we assume the low-fidelity model to be defined even outside the original support of input parameters, and we are currently investigating how to lift such assumption.

Similarly to the previous section, we compare our methods with standard (multifidelity) Monte Carlo techniques, by computing the mean value and the standard deviation of the estimators based on 100 samples, and plotting the corresponding Gaussian distribution. In the variance of the final distribution, we take into account all the possible sources of uncertainties, i.e., sampling from the input distribution and hyperparameter tuning. The optimal allocation is computed assuming a computational budget of 100 high-fidelity simulations.

The numerical results are displayed in Fig. 11 for both one-dimensional and two-dimensional shared spaces. We observe that our method based on active subspaces (MFMC AS) is not able to reduce the variance of the estimator. This implies either that a linear transformation is not enough to capture the directions where the models vary the most, or that the surrogate models are not sufficiently accurate to provide a good approximation of the gradient of the models, and consequently of the active subspaces. Alternative techniques could be explored to compute gradients, such as ridge regression [17] or adaptive basis [42]. Nevertheless, we do not consider these approaches here since the approximation of the gradient is not the main focus of this work. On the other hand, our technique based on autoencoders, and therefore nonlinear transformations, outperforms all the other approaches and provides a significant variance reduction while increasing the correlation between the high-fidelity and low-fidelity models. Moreover, we notice that in this case a one-dimensional shared space yields better results than a two-dimensional subspace. Future work will focus on how to determine the optimal dimensionality of the reduced space. This problem is more challenging for the method based on autoencoders because we do not have an order of importance provided by the eigenvalues as in the active subspace technique.

4.3 Cardiovascular simulation

We now consider a real application for which we have a limited data availability. We focus on cardiovascular blood flow simulations of coronary artery disease. In Fig. 12 we show the anatomic model together with the prescribed flow boundary condition at the aortic inlet and the intramyocardial pressures specified at the coronary outlets [19]. The

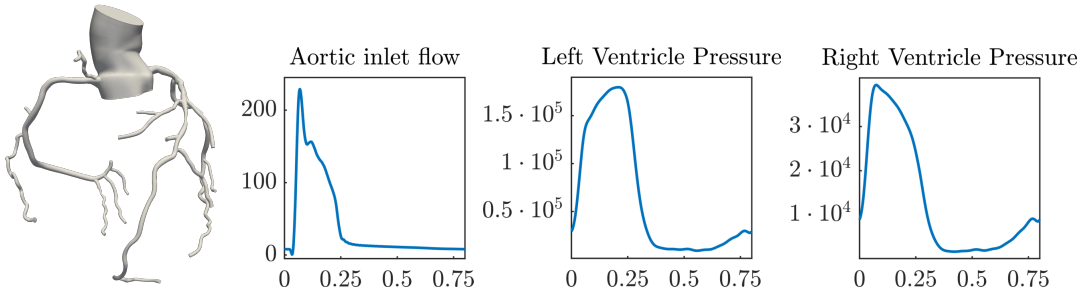

Figure 12: Coronary model for the cardiovascular simulations, together with the prescribed profiles for aortic inlet flow, left ventricle pressure, and right ventricle pressure.

Parameter	Mean Value	Unit	Std Dev
Total resistance	3193.5891	Barye/ml	50%
Total aorta capacitance	$1.6877 \cdot 10^{-4}$	ml/Barye	30%
Total left coronary arteries capacitance	$7.2127 \cdot 10^{-6}$	ml/Barye	30%
Total right coronary arteries capacitance	$7.2568 \cdot 10^{-6}$	ml/Barye	30%
Coronary Young's modulus	11500000.0	Barye	30%
Left coronary intramyocardial pressure	1.405165	Barye	20%
Right coronary intramyocardial pressure	4.394061	Barye	20%

Table 1: Input parameters and distribution of the cardiovascular simulations.

high-fidelity model is represented by 3D simulations, for which we generate two different low-fidelity models: 1D and 0D simulations. One-dimensional hemodynamics models [10] are formulated considering blood as a Newtonian fluid with velocity only along the axial direction of an ideal cylindrical branch, constant pressure over each vessel cross section, and a non-slip boundary condition at the vessel lumen. The governing equations are found by integrating the incompressible Navier-Stokes equations over the cross section of a deformable cylindrical domain, and the system of equations is completed with a constitutive model relating pressure to cross-sectional area deformations. On the other hand, zero-dimensional hemodynamics models are lumped parameter networks. They consist of an equivalent circuit model formulated by hydrodynamic analogy in terms of flow rate (electrical current) and pressure differences (voltage). Low fidelity models are created through an automated model generation pipeline recently implemented in SimVascular [30]. More details on the set up for the 3D simulations are provided in [26] and [25].

The input parameters of the models are listed in Table 1, and each parameter is distributed as a uniform random variable in the interval $\textit{Mean Value} \cdot (1 \pm \textit{Std Dev}/100)$, independent of the others. These values have been tuned so that model outputs correspond to physiological patient data. We sample 104 different parameters from the input distribution, and perform the corresponding 3D, 1D, and 0D simulations. We remark that each 3D simulation with deformable walls took approximately 15 hours on the Sherlock HPC cluster at Stanford using 4 nodes with 24 cores on an AMD EPYC 7502 processor with 2.5 GHz base CPU clock time and 32 GB of memory. We assume the cost ratios between the models to be $w^{1D} = 1.5 \cdot 10^{-3}$ and $w^{0D} = 2.5 \cdot 10^{-5}$.

We initially analyze several quantities of interest: minimum/maximum/average values in time of the outlet flow and pressure profile in the aorta and both the left and right coronary arteries, and minimum/maximum/average in space of the time averaged wall shear stresses in each coronary artery. We observe that in most of the cases the correlation

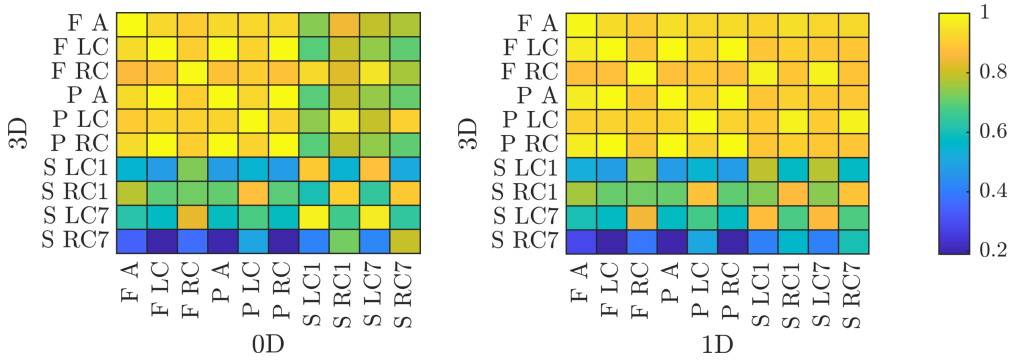

Figure 13: Matrix of the absolute value of the cross-correlation between 3D high-fidelity and 0D/1D low-fidelity cardiovascular simulations for the minimum value of different quantities of interest. The correlation between the same quantity of interest is represented on the diagonals. Notation: F flow, P pressure, S wall shear stress, A aorta, LC left coronary, RC right coronary.

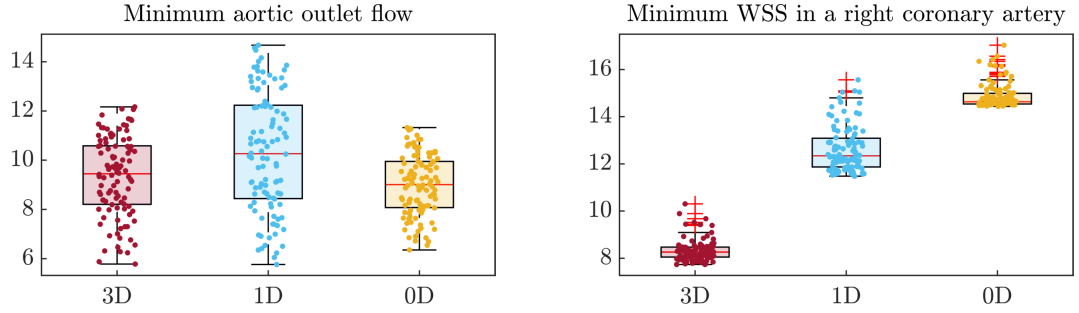

Figure 14: Boxplots of the quantities of interest computed from the cardiovascular simulations, employing 3D, 1D, and 0D models. Left: Minimum aortic outlet flow. Right: minimum wall shear stress computed in the right coronary artery with a stenosis.

between the 3D high-fidelity model and the 1D/0D low-fidelity models is extremely high, in particular for flow and pressures values. This implies that standard multifidelity Monte Carlo is already able to provide a significant variance reduction with respect to the Monte Carlo estimator. On the other hand, for wall shear stresses the correlation is notably smaller and this justifies the application of our methodologies. Indeed, wall shear stresses, being quantities strictly related to the 3D domain, are challenging to approximate through reduced order models. Moreover, since our approaches, in contrast to standard multifidelity Monte Carlo, create a shared parameterization between the two fidelities, we also compute the cross-correlation between the minimum value of different quantities, meaning that the 1D/0D simulation can measure a different output than the 3D simulation. In Fig. 13 we plot the matrices of the absolute value of the cross-correlations for some quantities of interest, and observe that the correlation can be poor for some pairs. We remark that if in concrete applications there is available data for which the two fidelities are poorly correlated, then our approaches can improve the estimation.

We then select the minimum wall shear stress in the right coronary artery which has a stenosis (S RC7 in Fig. 13) as the quantity of interest to approximate for all the 3D, 1D, and 0D results. Moreover, to showcase the versatility of our methodologies, we also consider a different quantity of interest for the low-fidelity model with respect to the high-fidelity. This opens the possibility to employ multifidelity estimation for any available data independently of the fact that they are related to the quantity under investigation. In

	0D			1D		
	Method	N^{HF}	N^{LF}	Method	N^{HF}	N^{LF}
Same QOI	MFMC	103	25721	MFMC	101	1870
	MFMC AE	103	44371	MFMC AE	99	3246
	Method	N^{HF}	N^{LF}	Method	N^{HF}	N^{LF}
Different QOI	MFMC	104	7133	MFMC	103	741
	MFMC AE	103	19297	MFMC AE	97	4044

Table 2: Optimal allocation for the multifidelity estimators for the cardiovascular simulations. The quantity measured by the low-fidelity model, both 0D (left) and 1D (right), is either the same (top) or different (bottom) from the quantity of interest computed by the 3D high-fidelity model.

	0D			1D		
	Method	Mean	Interval	Method	Mean	Interval
Same QOI	MC	8.332	±0.142	MC	8.332	±0.142
	MFMC	8.410	±0.079	MFMC	8.332	±0.114
	MFMC AE	8.381	±0.057	MFMC AE	8.390	±0.089
	Method	Mean	Interval	Method	Mean	Interval
Different QOI	MC	8.332	±0.142	MC	8.332	±0.142
	MFMC	8.331	±0.136	MFMC	8.339	±0.139
	MFMC AE	8.435	±0.101	MFMC AE	8.418	±0.079

Table 3: Values of the estimators (in Barye) together with their 90 % confidence interval for the cardiovascular simulations. The quantity measured by the low-fidelity model, both 0D (left) and 1D (right), is either the same (top) or different (bottom) from the quantity of interest computed by the 3D high-fidelity model.

particular, we then extract the minimum aortic outlet flow (F A in Fig. 13) as quantity of interest for the 1D and 0D simulations, which is readily available. In Fig. 14 we show the boxplots of the two quantities under consideration in this study. We observe that the aortic outlet flow can be better approximated by low-fidelity models compared to the wall shear stress.

We remark that, given the small number of simulations for 7-dimensional data, the standard active subspace approach introduced in this work does not provide better results than standard (multifidelity) Monte Carlo estimators. We therefore focus here on the particular choice of autoencoder described in Section 3.2.1, which only requires training a fully connected neural network surrogate for the 3D simulations, and two one-dimensional normalizing flows. The optimal allocation problems are solved assuming a computational budget of 104 simulations, in order to be able to compare the results with the standard Monte Carlo approach, and the resulting number of high-fidelity and low-fidelity simulations are rounded to the closest integer and given in Table 2. In Fig. 15 we show the results for the case when the quantity of interest computed by the low-fidelity model is the same quantity computed by the high-fidelity model, and in Fig. 16 when it is different. Moreover, in Table 3 we give the values with a 90 % confidence interval of the realizations of the

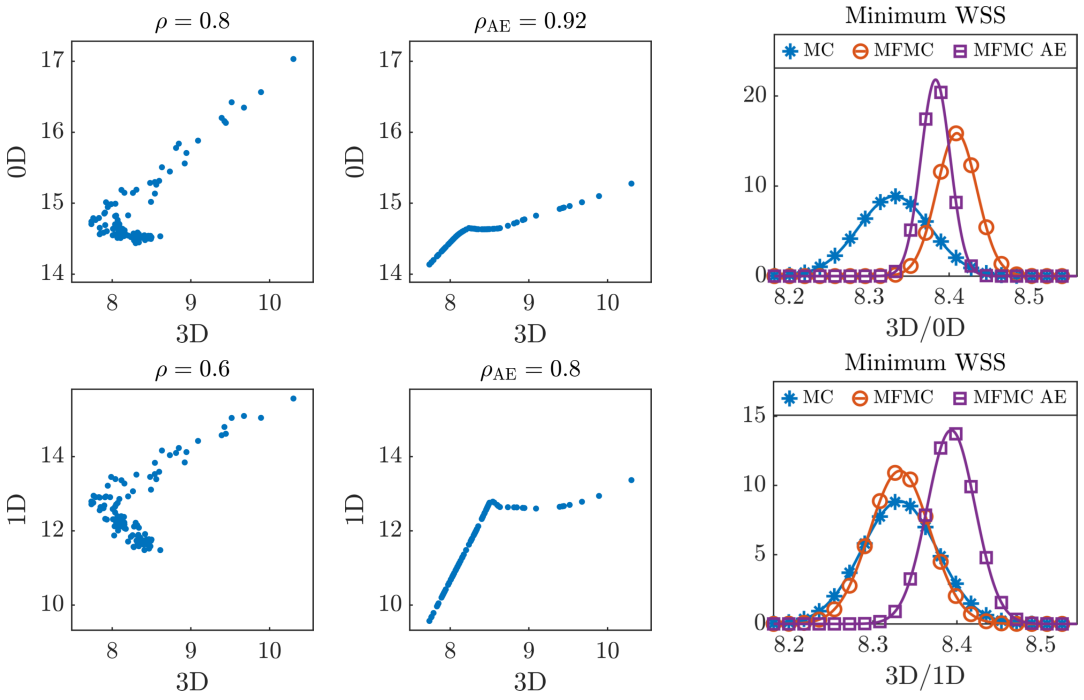

Figure 15: Comparison between MFMC AE with standard (multifidelity) Monte Carlo (MC and MFMC) for the cardiovascular simulations, where the quantity estimated by the high-fidelity and low-fidelity model is the same (minimum wall shear stress in the right coronary artery with a stenosis). Top: 0D low-fidelity model. Bottom: 1D low-fidelity model. Left: correlation between high-fidelity (3D) and low-fidelity data employing MFMC and MFMC AE. Right: estimated density distribution for the quantity of interest.

estimators. The intervals are obtained by multiplying the standard deviation given by equation (2.3) by $\sqrt{10}$ since due to the Chebyshev's inequality we have

$$\mathbb{P}(|\widehat{Q} - \mathbb{E}[Q(\boldsymbol{\xi})] \leq \sqrt{10}\sigma) \geq 0.9,$$

where $\widehat{Q}$ stands for any estimator and σ for its standard deviation. In both cases we first notice that the correlations of the new reduced low-fidelity models increase, and this implies a reduction in the variance of the estimators. The probability distributions in the last columns are indeed Gaussian distributions with mean given by the value of the estimator and variance computed from equation (2.3). We remark that the difference in estimation between multifidelity estimators with respect to standard Monte Carlo observed in the last column does not correspond to any bias. In fact, we just compute a single realization of the estimators and give a visual representation of these values, which are also reported in Table 3. In particular, these plots have to be interpreted differently from the similar plots in the previous sections. All the estimators are indeed asymptotically unbiased since we did not modify the high-fidelity model in their definition, and, if we could repeat the experiments multiple times, the average of the estimated values would give the exact mean of the quantity of interest, i.e., the minimum wall shear stress in the coronary artery with the stenosis. We finally notice that in Fig. 16 the fact that the quantities of interest of the low-fidelity and high-fidelity models are different does not affect the final results.

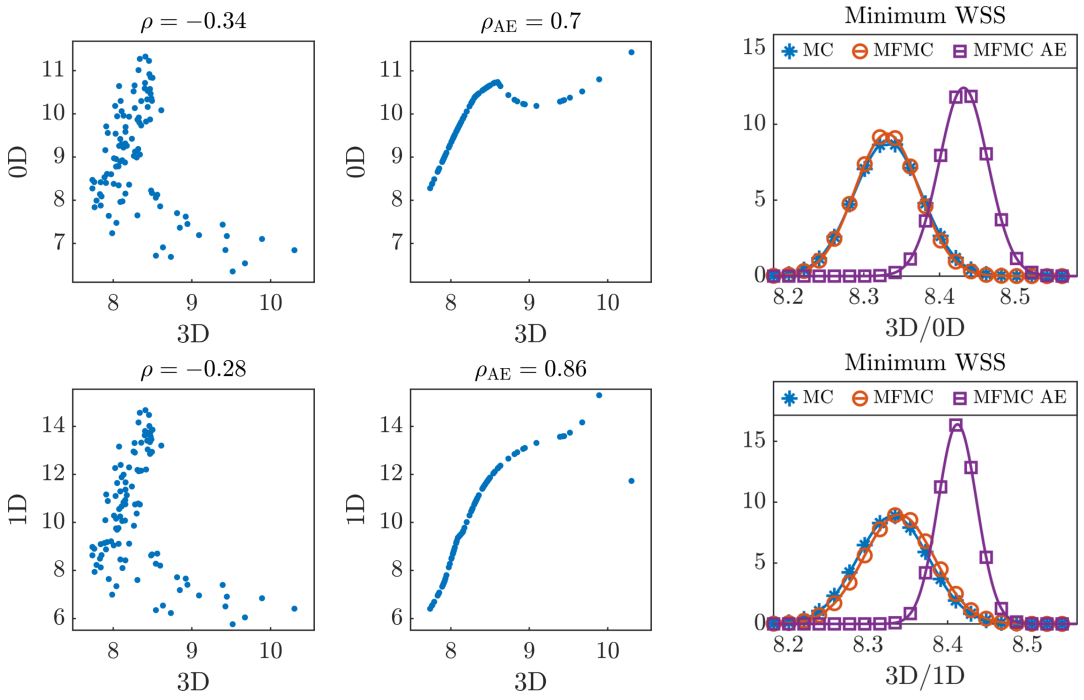

Figure 16: Comparison between MFMC AE with standard (multifidelity) Monte Carlo (MC and MFMC) for the cardiovascular simulations, where the quantity estimated by the high-fidelity and low-fidelity model is different (minimum wall shear stress in the right coronary artery with a stenosis for the high-fidelity model and minimum aortic outlet flow for the low-fidelity model). Top: 0D low-fidelity model. Bottom: 1D low-fidelity model. Left: correlation between high-fidelity (3D) and low-fidelity data employing MFMC and MFMC AE. Right: estimated density distribution for the quantity of interest.

5 Conclusions

In this work we proposed two different methodologies to improve multifidelity estimators for uncertainty propagation. In particular, we achieved variance reduction of the standard multifidelity Monte Carlo estimator by modifying the low-fidelity model in order to increase the correlation with the high-fidelity model. Our approaches rely on a shared space where the models vary the most and the parameters are distributed according to a standard Gaussian. We constructed the shared space through either linear or nonlinear dimensionality reduction techniques, namely active subspaces and autoencoders. We demonstrated by means of numerical experiments that, given sufficient data, the latter are able to find nonlinear transformations which allow us to further decrease the variance of the estimator with respect to linear transformations. Moreover, we employed normalizing flows to map different probability distributions into the same distribution, i.e., a standard Gaussian, and therefore generate a shared space. Our techniques not only permit getting an estimator with reduced variance, but also increase the range of applicability of multifidelity estimators. In particular, we allow for models with dissimilar parameterization, meaning that the number and type of input parameters between the high-fidelity and low-fidelity models and their distributions can be different. This implies that these approaches can also be applied to models which measure different quantities of interest which are not directly related, as long as the high-fidelity and the modified low-fidelity models are well correlated, as we showed in the challenging numerical examples involving cardiovascular simulations. A limitation

of our approach is that a large amount of data might be necessary to train the surrogate models, find the lower-dimensional subspaces, and build the normalizing flows. However, as we showed in Figs. 6 and 8 where we varied the number of data points, even if the best lower-dimensional manifolds are not correctly identified, we still have an improvement in terms of variance with respect to standard multifidelity Monte Carlo estimators. In other words, through the examples in the paper, we numerically show that the number of samples needed to build an accurate surrogate is typically much larger than those needed to build a surrogate for the sole purpose of identifying a shared space leading to a smaller variance than multifidelity Monte Carlo. The sensitivity analysis with respect to the number of samples also shows that, as expected, the variance of the resulting estimators decreases with a larger amount of data. Moreover, the construction of an accurate surrogate model could be lifted in the linear dimension reduction case by adopting strategies like adaptive basis (see, e.g., [45]) or it could be directly learned on the latent space together with the autoencoder. We leave this latter approach, which we expect to require a significantly smaller amount of data due to the reduced dimension, for future work. Another interesting extension of the current method is to train the autoencoders of both the high-fidelity and low-fidelity models together, and include a term in the loss function that maximizes the resulting correlation and therefore improves the variance of the estimators. Finally, we are also interested in studying a method to automatically find the optimal reduced dimension of the shared space, which is fundamental for applications with high-dimensional input parameters.

Acknowledgements

This work is supported by NSF CAREER award #1942662 (DES), NSF CDS&E award #2104831 (DES), NSF award #2105345 (ALM), and NIH grant #R01HL141712 (ALM, KM). 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. The authors thank Boris Kramer for insightful suggestions about Section 3.3.

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] 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.

[4] P. Constantine, Q. Wang, A. Doostan, and G. Iaccarino, *A Surrogate Accelerated Bayesian Inverse Analysis of the HyShot II Flight Data*, 4 2011, ch. paper AIAA-2011-2037.

[5] 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.

[6] M. Croci, K. Willcox, and S. Wright, *Multi-output multilevel best linear unbiased estimators via semidefinite programming*, Computer Methods in Applied Mechanics and Engineering, 413 (2023), p. 116130.

[7] L. Dinh, J. Sohl-Dickstein, and S. Bengio, *Density estimation using real NVP*, in International Conference on Learning Representations, 2017.

[8] V. G. Eck, W. P. Donders, J. Sturdy, J. Feinberg, T. Delhaas, L. R. Hellevik, and W. Huberts, *A guide to uncertainty quantification and sensitivity analysis for cardiovascular applications*, International Journal for Numerical Methods in Biomedical Engineering, 32 (2015).

[9] C. M. Fleeter, G. Geraci, D. E. Schiavazzi, A. M. Kahn, and A. L. Marsden, *Multilevel and multifidelity uncertainty quantification for cardiovascular hemodynamics*, Computer Methods in Applied Mechanics and Engineering, 365 (2020), p. 113030.

[10] L. Formaggia, F. Nobile, A. Quarteroni, and A. Veneziani, *Multiscale modelling of the circulatory system: A preliminary analysis*, Computing and Visualization in Science, 2 (1999), pp. 75–83.

[11] G. Geraci and M. Eldred, *Leveraging intrinsic principal directions for multifidelity uncertainty quantification*, SAND2018-10817, (2018).

[12] 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.

[13] G. Geraci, M. S. Eldred, A. Gorodetsky, and J. Jakeman, *Recent advancements in Multilevel-Multifidelity techniques for forward UQ in the DARPA Sequoia project.*

[14] M. B. Giles, *Multi-level monte carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.

[15] 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.

[16] G. Granato, *Stochastic empirical loading and dilution model (SELDM) version 1.0.0*, 03 2013.

[17] J. M. Hokanson and P. G. Constantine, *Data-driven polynomial ridge approximation using variable projection*, SIAM J. Sci. Comput., 40 (2018), pp. A1566–A1589.

[18] T. J. Hughes and J. Lubliner, *On the one-dimensional theory of blood flow in the larger vessels*, Math Biosci, 18 (1973), pp. 161–170.

[19] H. J. Kim, I. E. Vignon-Clementel, J. S. Coogan, C. A. Figueroa, K. E. Jansen, and C. A. Taylor, *Patient-specific modeling of blood flow and pressure in human coronary arteries*, Annals of Biomedical Engineering, 38 (2010), pp. 3195–3209.

[20] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017.

[21] G. A. Klaasen and W. C. Troy, *Stationary wave solutions of a system of reaction-diffusion equations derived from the FitzHugh-Nagumo equations*, SIAM J. Appl. Math., 44 (1984), pp. 96–110.

[22] 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.

[23] R. R. Lam, O. Zahm, Y. M. Marzouk, and K. E. Willcox, *Multifidelity dimension reduction via active subspaces*, SIAM Journal on Scientific Computing, 42 (2020), pp. A929–A956.

[24] Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini, *Sampling via Measure Transport: An Introduction*, Springer International Publishing, Cham, 2016, pp. 1–41.

[25] K. Menon, M. O. Khan, Z. A. Sexton, J. Richter, K. Nieman, and A. L. Marsden, *Personalized coronary and myocardial blood flow models incorporating ct perfusion imaging and synthetic vascular trees*, medRxiv, 2023.08.17.23294242 (2023).

[26] K. Menon, J. Seo, R. Fukazawa, S. Ogawa, A. M. Kahn, J. C. Burns, and A. L. Marsden, *Predictors of myocardial ischemia in patients with kawasaki disease: Insights from patient-specific simulations of coronary hemodynamics*, Journal of Cardiovascular Translational Research, 16 (2023), p. 1099–1109.

[27] L. W. T. Ng and K. E. Willcox, *Multifidelity approaches for optimization under uncertainty*, Internat. J. Numer. Methods Engrg., 100 (2014), pp. 746–772.

[28] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, J. Mach. Learn. Res., 22 (2021), pp. Paper No. 57, 64.

[29] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Optimal model management for multifidelity Monte Carlo estimation*, SIAM J. Sci. Comput., 38 (2016), pp. A3163–A3194.

[30] M. R. Pfaller, J. Pham, A. Verma, L. Pegolotti, N. M. Wilson, D. W. Parker, W. Yang, and A. L. Marsden, *Automated generation of 0d and 1d reduced-order models of patient-specific blood flow*, International Journal for Numerical Methods in Biomedical Engineering, n/a (2022), p. e3639.

[31] A. Quarteroni, S. Ragni, and A. Veneziani, *Coupling between lumped and distributed models for blood flow problems*, Comput Vis Sci, 4 (2001), pp. 111–124.

[32] F. Regazzoni, M. Salvador, P. Africa, M. Fedele, L. Dedè, 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.

[33] M. Salvador, F. Regazzoni, L. Dede', and A. Quarteroni, *Fast and robust parameter estimation with uncertainty quantification for the cardiac function*, Computer Methods and Programs in Biomedicine, 231 (2023), p. 107402.

[34] D. Schaden and E. Ullmann, *On multilevel best linear unbiased estimators*, SIAM/ASA Journal on Uncertainty Quantification, 8 (2020), pp. 601–635.

[35] D. Schaden and E. Ullmann, *Asymptotic analysis of multilevel best linear unbiased estimators*, SIAM/ASA Journal on Uncertainty Quantification, 9 (2021), pp. 953–978.

[36] J. Seo, C. Fleeter, A. M. Kahn, A. L. Marsden, and D. E. Schiavazzi, *Multifidelity estimators for coronary circulation models under clinically informed data uncertainty*, International Journal for Uncertainty Quantification, 10 (2020), pp. 449–466.

[37] J. Seo, D. E. Schiavazzi, A. M. Kahn, and A. L. Marsden, *The effects of clinically-derived parametric data uncertainty in patient-specific coronary simulations with deformable walls*, International Journal for Numerical Methods in Biomedical Engineering, 36 (2020), p. e3351.

[38] A. Spantini, D. Bigoni, and Y. Marzouk, *Inference via low-dimensional couplings*, J. Mach. Learn. Res., 19 (2018), p. 2639–2709.

[39] D. A. Steinman and F. Migliavacca, *Editorial: Special issue on verification, validation, and uncertainty quantification of cardiovascular models: Towards effective VVUQ for translating cardiovascular modelling to clinical utilitys*, Cardiovascular Engineering and Technology, 9 (2018).

[40] V. Stimper, D. Liu, A. Campbell, V. Berenz, L. Ryll, B. Schölkopf, and J. M. Hernández-Lobato, *normflows: A pytorch package for normalizing flows*, Journal of Open Source Software, 8 (2023), p. 5361.

[41] M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert, *Pdebench: An extensive benchmark for scientific machine learning*, 2023.

[42] R. Tipireddy and R. Ghanem, *Basis adaptation in homogeneous chaos spaces*, J. Comput. Phys., 259 (2014), pp. 304–317.

[43] A. M. Turing, *The chemical basis of morphogenesis*, Philos. Trans. Roy. Soc. London Ser. B, 237 (1952), pp. 37–72.

[44] Y. Wang, F. Liu, and D. E. Schiavazzi, *Variational inference with nofas: Normalizing flow with adaptive surrogate for computationally expensive models*, Journal of Computational Physics, 467 (2022), p. 111454.

[45] 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.