# A survey on multi-fidelity surrogates for simulators with functional outputs: unified framework and benchmark

Lucas Brunel, Mathieu Balesdent, Loïc Brevault, Rodolphe Le Riche, Bruno Sudret

ETH zürich

# A SURVEY ON MULTI-FIDELITY SURROGATES FOR SIMULATORS WITH FUNCTIONAL OUTPUTS: UNIFIED FRAMEWORK AND BENCHMARK

L. Brunel, M. Balesdent, L. Brévault, R. Le Riche and B. Sudret


CHAIR OF RISK, SAFETY AND UNCERTAINTY QUANTIFICATION
STEFANO-FRANSCINI-PLATZ 5
CH-8093 ZÜRICH

Risk, Safety & Uncertainty Quantification

Data Sheet


| | |
|---|---|
| **Journal:** | Computer Methods in Applied Mechanics and Engineering |
| **Report Ref.:** | RSUQ-2024-006A |
| **Arxiv Ref.:** | https://arxiv.org/abs/2408.17075 [cs.CE] |
| **DOI:** | - |
| **Date submitted:** | August 30, 2024 |
| **Date accepted:** | - |

# A survey on multi-fidelity surrogates for simulators with functional outputs: unified framework and benchmark

Lucas Brunel[a,b], Mathieu Balesdent[a], Loïc Brevault[a], Rodolphe Le Riche[b], Bruno Sudret[c]

[a] *ONERA, Paris Saclay University, Palaiseau, F-91123, France*
[b] *LIMOS (CNRS, Mines Saint-Etienne and Université Clermont Auvergne), France*
[c] *ETH Zürich, Chair of Risk, Safety and Uncertainty Quantification, Switzerland*

**Abstract**

Multi-fidelity surrogate models combining dimensionality reduction and an intermediate surrogate in the reduced space allow a cost-effective emulation of simulators with functional outputs. The surrogate is an input-output mapping learned from a limited number of simulator evaluations. This computational efficiency makes surrogates commonly used for many-query tasks. Diverse methods for building them have been proposed in the literature, but they have only been partially compared.

This paper introduces a unified framework encompassing the different surrogate families, followed by a methodological comparison and the exposition of practical considerations. More than a dozen of existing multi-fidelity surrogates have been implemented under the unified framework and evaluated on a set of benchmark problems. Based on the results, guidelines and recommendations are proposed regarding multi-fidelity surrogates with functional outputs.

Our study shows that most multi-fidelity surrogates outperform their tested single-fidelity counterparts under the considered settings. But no particular surrogate is performing better on every test case. Therefore, the selection of a surrogate should consider the specific properties of the emulated functions, in particular the correlation between the low- and high-fidelity simulators, the size of the training set, the local nonlinear variations in the residual fields, and the size of the training datasets.

*Keywords:* surrogate modeling, multi-fidelity, functional outputs

## 1. Introduction

In computational science and engineering, many-query tasks such as optimization, uncertainty quantification, and sensitivity analysis play crucial roles in decision-making processes. These tasks involve numerous simulations of complex phenomena. In classical engineering scenarios, the simulator output is functional (*e.g.*, a scalar field) and discretized on a high-dimensional mesh, which can, for instance, be temporal or spatial. The computational cost of a single run of a complex simulator can be substantial, making a large number of evaluations unaffordable in practice. A common solution to this challenge is the construction of a mathematical approximation of the simulator, known as a *surrogate model*, or simply *surrogate*. Surrogate models are significantly cheaper to evaluate than the original simulators, enabling cost-effective predictions of the simulator outputs. In most cases, multiple fidelities of simulator are available, by varying the level of simplifications in the physics of the problem, the mesh refinement, the convergence tolerance, *etc.* For instance, in Computational Fluid Dynamics (CFD), a high-fidelity model could be a CFD Reynolds-Averaged Navier-Stokes (RANS) simulator on a fine mesh, while a low-fidelity one could be a simpler CFD Euler simulator on a coarse mesh. The surrogate can be adapted to exploit the multiple fidelities available, thus becoming a *multi-fidelity* surrogate.

Among the existing approaches for constructing multi-fidelity surrogates capable of predicting functional outputs, two main families can be distinguished. The first combines two separate tasks: dimensionality reduction (DR) and intermediate surrogate modeling [1–12]. Taking advantage of spatial or temporal dependence of the nodes of the output field, DR describes high-dimensional data by a significantly smaller number of variables, providing a mapping between the original high-dimensional space and the new low-dimensional space. Then, intermediate surrogate modeling is performed between the simulator input space and this new low-dimensional space on this limited number of variables. The second family of approaches intertwines DR and intermediate surrogate modeling into a unified task, thanks to artificial neural networks (see, *e.g.*, [13–15]).

Although several multi-fidelity surrogates of the first family (*i.e.*, combining DR and intermediate surrogate modeling) have been proposed in recent literature [1–12], there has been limited theoretical and numerical comparison between them. Moreover, each multi-fidelity surrogate has been described in its own particular formalism. This paper aims at describing these existing

multi-fidelity surrogates with functional outputs within a *unified framework* and at benchmarking them on a set of test cases. The proposed framework is designed to be modular. It facilitates the adaptation of existing surrogates and the development of new ones. The benchmark involves viscous flow and airfoil design test cases of increasing complexity, considering two simulator fidelities, and is followed by thorough discussions. Recommendations based on the theoretical and numerical comparisons are provided to assist practitioners in choosing the appropriate surrogate for a given application.

The outline of this paper is as follows. Section 2 describes the different building blocks of the surrogates in a unified formalism. Section 3 presents the different families of multi-fidelity surrogates and briefly details each of them. Section 4 introduces the test cases making up the benchmark. Section 5 presents the benchmark methodology and the comparison metrics. Finally, Section 6 displays and discusses the performance of the surrogates on the test cases to provide practitioners with guidelines depending on the characteristics of their problems.

## 2. Background

In this work, several simulators with hierarchical fidelities are considered. The difference in fidelity may be based on different aspects such as physical simplifications (*e.g.*, CFD Large Eddy Simulation (LES) for the high-fidelity and CFD Euler for the low-fidelity, which simplifies the modeling of viscosity), numerical discretization (*e.g.*, fine and coarse mesh) or numerical tolerances (*e.g.*, different thresholds of residuals involved in the solvers). The fidelities can be described by their computational cost and accuracy. A low-fidelity simulator presents a reduced computational cost but a low accuracy, while a high-fidelity simulator is more accurate but has a large computational cost. The level of fidelity is denoted by the index $l \in \{1, \dots, L\}$, sorted in decreasing fidelity order such that $l = 1$ corresponds to the highest fidelity and $l = L$ to the lowest fidelity. It is assumed that the input variables of these simulators (*e.g.*, design or environment variables) are the same for each fidelity and are gathered in the vector $\mathbf{u} \in U \subseteq \mathbb{R}^{d_\mathbf{u}}$. Each simulator $\mathcal{S}_l$ of fidelity level $l$ outputs a scalar valued field $\mathbf{y}_l \in Y_l \subseteq \mathbb{R}^{d_{\mathbf{y}_l}}$ that corresponds to a high-dimensional mesh $\mathbf{X}_l \in X_l \subseteq \mathbb{R}^{d_{\mathbf{y}_l} \times d_\mathbf{x}}$ (*e.g.*, spatial or temporal), with $d_{\mathbf{y}_l}$ the number of nodes, or vertices, in a $d_\mathbf{x}$-dimensional space, corresponding to the mesh coordinates in the associated space. $d_\mathbf{x}$ is assumed to be the same for all fidelities (*e.g.*, all simulators output two-dimensional fields).

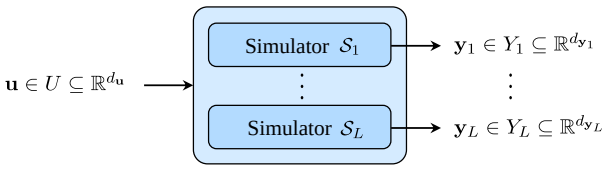



Figure 1: Sketch of a generic multi-fidelity simulation, with the input vector $\mathbf{u} \in U$ and at fidelity level $l \in \{1, \dots, L\}$ the simulator $\mathcal{S}_l$, the mesh $\mathbf{X}_l \in X_l$ and the scalar output field $\mathbf{y}_l \in Y_l$.

The $j$-th component of the output field is the scalar $y_{j,l} \in \mathbb{R}$, with $\mathbf{x}_{j,l}$ the coordinate of the $j$-th node of $\mathbf{X}_l$. Note that fidelities can have a different number of mesh nodes $d_{\mathbf{y}_l}$, or the same $\forall l \in \{1, \dots, L\}, d_{\mathbf{y}_l} = d_\mathbf{y}$ but with coordinates that may vary between snapshots (see [16] for illustrations). To summarize, the simulator $\mathcal{S}_l$ of fidelity level $l$ discretized on the mesh $\mathbf{X}_l$ is the mapping

$$\begin{aligned} \mathcal{S}_l : U &\to Y_l \\ \mathbf{u} &\mapsto \mathbf{y}_l = \mathcal{S}_l(\mathbf{u}; \mathbf{X}_l). \end{aligned} \tag{1}$$

Whenever the reference to the mesh $\mathbf{X}_l$ is not necessary, the above-mentioned equation is simply written as $\mathbf{y}_l = \mathcal{S}_l(\mathbf{u})$. The multi-fidelity simulation setup described is illustrated in Figure 1.

### *2.1. Overview of DR techniques*

#### *2.1.1. Single-fidelity techniques*

DR is first introduced for the case of a single fidelity level $l$. In this work, $Y_l$ is supposed to be a vector space. The underlying physical model of the simulator implies that the components of the output field are usually (spatially, temporally, *etc.*) dependent. DR assumes that the observed discretized output fields lay on or near a subspace $Z_l$ of the high-dimensional vector space $Y_l$. This subspace is called the *latent space* and is mathematically modeled by a $d_{\mathbf{z}_l}$-dimensional manifold. The coordinates on the manifold are the *latent variables* and denoted by $\mathbf{z}_l$. $d_{\mathbf{z}_l}$ is the *intrinsic dimension* [17] and $d_{\mathbf{z}_l} < d_{\mathbf{y}_l}$ (often $d_{\mathbf{z}_l} \ll d_{\mathbf{y}_l}$). Usually, neither the dimensionality $d_{\mathbf{z}_l}$ nor the structure of the manifold (differentiable, discontinuous, *etc.*) is known. Performing DR consists in finding the mapping $\mathcal{DR}_l^{\boldsymbol{\omega}_l}$ from the high-dimensional vector

space $Y_l$ to the lower-dimensional space $Z_l$, with $\boldsymbol{\omega}_l \in \mathbb{R}^{d_{\boldsymbol{\omega}_l}}$ a set of hyperparameters which the mapping may depend on. This mapping is summarized as:

$$\begin{aligned} \mathcal{DR}_l^{\boldsymbol{\omega}_l} : Y_l &\to Z_l \\ \mathbf{y}_l &\mapsto \mathbf{z}_l \end{aligned} \tag{2}$$

For the sake of readability, the dependence of $\mathcal{DR}_l^{\boldsymbol{\omega}_l}$ on $\boldsymbol{\omega}_l$ is omitted in the following. $\mathbf{z}_l$ is then a low-dimensional representation of the output field $\mathbf{y}_l$. In practice, assumptions on its structure and dimensionality are required in order to be able to work with $Z_l$ and to compute $\mathcal{DR}_l^{\boldsymbol{\omega}_l}$. Ideally, this transformation should not yield information loss. In practice though, since assumptions are made on the structure and dimensionality of the manifold, some information may be lost.

DR techniques are mostly data-driven. In order to be accurate enough, they require a certain amount of *snapshots*, *i.e.*, a set of output fields of the simulator $\mathcal{S}_l$ obtained for different values of the input variables. The $n_l$ samples of $\mathbf{u}$ for which the simulator is evaluated are collected into the input Design of Experiments (DoE) $\mathcal{U}_l = \left\{\mathbf{u}_l^{(i)}\right\}_{i=1}^{n_l}$, with $\mathbf{I}_l = \left[\mathbf{u}_l^{(1)}, \dots, \mathbf{u}_l^{(n_l)}\right]^\top$ its associated $n_l$-by-$d_{\mathbf{u}}$ matrix form. The output DoE at fidelity $l$ is denoted by $\mathcal{Y}_l = \left\{\mathbf{y}_l^{(i)}\right\}_{i=1}^{n_l}$ and its $n_l$-by-$d_{\mathbf{y}_l}$ matrix form $\mathbf{O}_l = \left[\mathbf{y}_l^{(1)}, \dots, \mathbf{y}_l^{(n_l)}\right]^\top$. The yet to be computed collection of the latent variables corresponding to all the snapshots of the output DoE is $\mathcal{Z}_l = \left\{\mathbf{z}_l^{(i)}\right\}_{i=1}^{n_l}$ and its $n_l$-by-$d_{\mathbf{z}_l}$ matrix form $\mathbf{L}_l = \left[\mathbf{z}_l^{(1)}, \dots, \mathbf{z}_l^{(n_l)}\right]^\top$. The strategy for choosing these samples may have a large impact on the quality of the DR. A common sampling method is Latin Hypercube Sampling (LHS) [18], designed to cover the input parameter space with uniform marginal distributions.

There exists a wide range of DR techniques that have been developed for different fields of research and engineering. Multiple classifications of these techniques have been proposed, such as [19–21].

Dimensionality reduction techniques can be classified according to the linearity of their mapping. Linear techniques are based on the hypothesis that the manifold is a vector space and the mapping between the original and the latent spaces is linear. The most well-known linear technique is *Principal Component Analysis* (PCA), also known as proper orthogonal decomposition

in the model order reduction community, and Karhunen-Loève expansion for stochastic processes [22]. As the scalar field is discretized, it is based on a decomposition of the covariance matrix associated with snapshots over an orthonormal basis, consisting in a set of eigenvectors and eigenvalues (Mercer's theorem [23]). In practice, the basis is truncated by retaining the $d_{\mathbf{z}_l}$ eigenvectors corresponding to the largest eigenvalues. The dimensionality $d_{\mathbf{z}_l}$ of the latent space is considered to be a hyperparameter of PCA as it is not known beforehand and has to be tuned. It is often chosen based on the Relative Information Content (RIC) [24] defined as the sum of the $d_{\mathbf{z}_l}$ largest eigenvalues divided by the sum of all eigenvalues. Factor Analysis [25] and Independent Component Analysis [26] are other examples of linear DR methods.

Nonlinear techniques offer more flexibility for the structure of the manifold. However, some lack a closed-form inverse mapping $\mathcal{DR}_l^{-1}$ from the latent space to the original high-dimensional space. They involve hyperparameters in addition to the dimensionality of the latent space that have to be chosen as well. A first example of nonlinear DR technique is Kernel PCA [27], which stems from PCA, combined with the kernel trick [28]. The snapshots of dimensionality $d_{\mathbf{y}_l}$ are nonlinearly mapped to an even higher-dimensional feature space, where PCA is performed. Consequently, the mapping between the original space $Y_l$ and the latent space $Z_l$ is nonlinear. The kernel is chosen such that no computation is actually done in the higher-dimensional space. The correlation length of the kernel is another hyperparameter that needs tuning. The family of techniques referred to as manifold learning are also used for nonlinear DR techniques (*e.g.*, Isomap [29], Local Tangent Space Alignment (LTSA) [30]). Autoencoders are another notable nonlinear technique [31], which rely on artificial neural networks to compute the low-dimensional representation of the snapshots.

*2.1.2. Multi-fidelity techniques*

The extension of DR to the multi-fidelity case can be accomplished in three different ways. The first consists in computing the DR mapping for one of the fidelity levels $\tilde{l}$ and to use this mapping $\mathcal{DR}_{\tilde{l}}$ to compute the low-dimensional representation of the snapshots for all fidelity levels (*i.e.*, $\mathcal{DR}_l = \mathcal{DR}_{\tilde{l}}, \forall l \in \{1, \ldots, L\}$) [3, 8]. All the output fields of the different fidelity levels are consequently mapped to the same manifold $Z_{\tilde{l}}$. For this purpose, every output field must be discretized on the same mesh. Otherwise, a preprocessing step must be applied to project every snapshots onto

a common mesh. It also assumes that DR computed at fidelity level $\tilde{l}$ can accurately represent snapshots of all fidelity levels, *i.e.*, that the physical information captured at fidelity level $\tilde{l}$ also describes the physics of other fidelity snapshots. This approach makes the use of the DR techniques described in Section 2.1.1 possible. The choice of the fidelity level $\tilde{l}$ is of importance and may be driven by the amount of snapshots at certain fidelity levels [3]. For instance, if too few higher-fidelity snapshots are available, one may use a lower-fidelity to perform DR.

The second way is to apply a DR technique designed to be able to use the aggregated snapshots available for all fidelity levels (*i.e.*, $\mathcal{DR}_l = \mathcal{DR}, \forall l \in \{1, \ldots, L\}$, with $\mathcal{DR}$ computed using all snapshots at once) [1, 7, 9, 11]. Thus, snapshots of all fidelity levels are mapped to the same shared manifold. This approach requires the definition of new DR techniques, or the adaptation of the ones introduced in Section 2.1.1. In most cases, this requires the snapshots of all fidelity levels to be discretized on the same mesh (*e.g.*, in PCA to compute the covariance between each component of the discretized field, in Isomap to compute the distance between the different snapshots). For instance, in PCA, this can be accomplished by computing the eigenvectors of the covariance matrix of the union of the snapshots of all fidelity levels [7, 11].

The third way is to independently perform one DR per fidelity (*i.e.*, $\mathcal{DR}_l \neq \mathcal{DR}_{l'}, \forall l \neq l' \in \{1, \ldots, L\}$) [2, 5, 6, 10]. Consequently, there are as many DR mappings as fidelity levels. To be able to transfer information between the different manifolds $Z_l$, this approach requires that either manifolds have a similar structure at every fidelity level, or a mapping between the different manifolds can be found, such as *manifold alignment* presented in Section 2.2. Again, this kind of approaches is compatible with the DR techniques described in Section 2.1.1.

### *2.2. Manifold alignment*

Manifold alignment is a method that provides a mapping between distinct low-dimensional representation of datasets that are associated to different manifolds. There are two main approaches to do so.

In the first approach, the DR with different fidelity snapshots is carried out such that a common manifold $Z$ for all distinct datasets is found [32, 33]. The DR technique has to be adapted so that each mapping $\mathcal{DR}_l$ defines a transformation from the original space $Y_l$ at a fidelity $l$ to the common manifold $Z$ ($\mathcal{DR}_l : Y_l \to Z, \forall l \in \{1, \ldots, L\}$). This approach can be achieved

either with a set of correspondences between some input samples from the distinct datasets [32], or without [33]. In the context of this paper, a set of correspondence means that some snapshots have been obtained for the same input variables values for different fidelity levels $l$ and $l'$, *i.e.*, there are some $\mathbf{u}_l^{(i)}, i = 1, \ldots, n_c$, with $n_c$ the number of common input variables vectors, such that $\mathbf{u}_l^{(i)} \in \mathcal{U}_l$ and $\mathbf{u}_l^{(i)} \in \mathcal{U}_{l'}$.

In the second approach, a separate DR is performed for each fidelity, leading to $L$ mappings $\mathcal{DR}_l$ and as many manifolds $Z_l$. The goal is then to find a transformation that maps one manifold to another. It can be achieved through an orthogonal *Procrustes analysis* [34]. It is based on the minimization of the Frobenius norm between the matrices $\mathbf{L}_l$, assuming that the manifolds all have the same dimensionality $d_{\mathbf{z}_l} = d_{\mathbf{z}}, \forall l \in \{1, \ldots, L\}$ [35]. Note that this method requires some correspondence between the input variables of some samples from the different fidelity models. It defines a set of affine transformations (scaling, translation and generalized rotation) applied to a given manifold in order to match a target manifold. Manifold alignment by orthogonal Procrustes analysis between the fidelity levels $l$ and $l'$ is then the mapping

$$\begin{aligned} \mathcal{MA} : \mathbb{R}^{d_{\mathbf{z}}} &\to \mathbb{R}^{d_{\mathbf{z}}} \\ \mathbf{z}_l &\mapsto \mathbf{z}_{l'}. \end{aligned} \tag{3}$$

If the dimensionalities of the manifolds are different, manifold alignment by orthogonal Procrustes analysis can be extended to projection Procrustes analysis [36].

### *2.3. Modeling the DR residuals by Gaussian processes with tensorized covariance*

The DR residuals correspond to the part of information that is lost when performing DR. It is the difference between the original snapshots and the snapshots projected onto the latent space and back-projected to the original space. These residuals computed for every snapshots, $\mathcal{Y}_{l,\perp} = \left\{\mathbf{y}_{l,\perp}^{(i)}\right\}_{i=1}^{n_l}$, are as follows:

$$\mathbf{y}_{l,\perp}^{(i)} = \mathbf{y}_l^{(i)} - \mathcal{DR}_l^{-1}\left(\mathcal{DR}_l\left(\mathbf{y}_l^{(i)}\right)\right), i = 1, \ldots, n_l. \tag{4}$$

Notice that $\mathcal{DR}_l^{-1}$ is not necessarily the exact inverse mapping of $\mathcal{DR}_l$ but a numerical approximation thereof. In Kerleguer [3], it is referred to as

the *orthogonal part*. The name "orthogonal" comes from the fact that in PCA, when projecting onto the reduced basis, the DR residuals correspond to the elements projected onto the space orthogonal to the one spanned by the PCA basis vectors. These residuals may be modeled by any single-fidelity surrogate with functional outputs. Among others, the approach introduced in [3] is summarized in the following.

Gaussian process (GP) regression with tensorized covariance [37, 38] is a single-fidelity surrogate model with functional outputs. The DR residuals are modeled by a GP which depends on the input vector $\mathbf{u} \in U$ and the location (spatial, temporal, *etc.*) in the field, denoted by $\mathbf{x}_{j,l} \in \mathbb{R}^{d_\mathbf{x}}$, $j \in \{1, \ldots, d_{\mathbf{y}_l}\}$. Let $\mathcal{S}_{l,\perp}^{\mathbf{x}_{j,l}}(\mathbf{u})$ be the value of the DR residuals at location $\mathbf{x}_{j,l}$. The covariance of this GP for two different input samples $\mathbf{u}$ and $\mathbf{u}'$ and two different locations $\mathbf{x}_{j,l}$ and $\mathbf{x}_{k,l}$ is modeled by the covariance function (also called a *kernel*) $\phi(\cdot)$ such that

$$\operatorname{cov}\left(\mathcal{S}_{l,\perp}^{\mathbf{x}_{j,l}}(\mathbf{u}), \mathcal{S}_{l,\perp}^{\mathbf{x}_{k,l}}(\mathbf{u}')\right) = \phi\left(\mathbf{u}, \mathbf{u}', \mathbf{x}_{j,l}, \mathbf{x}_{k,l}\right). \tag{5}$$

Since the product of covariance functions is a valid covariance function, it is further assumed that this covariance structure can be separated into a covariance function for the input variables $\phi_\mathbf{u}(\cdot)$ and a covariance function for the location in the field $\phi_\mathbf{x}(\cdot)$, such that

$$\operatorname{cov}\left(\mathcal{S}_{l,\perp}^{\mathbf{x}_{j,l}}(\mathbf{u}), \mathcal{S}_{l,\perp}^{\mathbf{x}_{k,l}}(\mathbf{u}')\right) = \phi_\mathbf{u}(\mathbf{u}, \mathbf{u}') \cdot \phi_\mathbf{x}\left(\mathbf{x}_{j,l}, \mathbf{x}_{k,l}\right). \tag{6}$$

If the field is discretized on a mesh that does not vary with respect to the input vector, $\phi_\mathbf{x}(\cdot)$ can be replaced by a $d_{\mathbf{y}_l}$-by-$d_{\mathbf{y}_l}$ covariance matrix $\mathbf{\Phi}_\mathbf{x}$. Moreover, if the simulator has been evaluated on a set of $n_l$ input vectors, the $n_l$-by-$n_l$ covariance in input vector matrix $\mathbf{\Phi}_\mathbf{u}$ can be computed. This is obtained via a classical covariance kernel, possibly with hyperparameters that have to be tuned. The covariance kernel $\mathbf{\Phi}$ of the GP is then obtained as the tensor product

$$\mathbf{\Phi} = \mathbf{\Phi}_\mathbf{u} \otimes \mathbf{\Phi}_\mathbf{x}. \tag{7}$$

This type of single-fidelity surrogate model can be used to model any simulator with functional output. It is used in the following as a single-fidelity surrogate for comparison purposes.

### *2.4. Intermediate surrogate modeling approaches*

Surrogate models are mathematical approximations of functions whose output can only be known for a limited set of input variables values. They

can be used to replace computationally expensive functions or in the context of this study, to create a mapping between the input variables $\mathbf{u} \in U$ and the latent space variables $\mathbf{z}_l \in Z_l$. The prediction accuracy depends on different factors such as the number and distribution of samples within the input space $U$, the properties of the unknown function (*e.g.*, continuity, stationarity, smoothness) or the modeling hypotheses (*e.g.*, number of coefficients for a polynomial regression, covariance model for a Gaussian process). They also often involve hyperparameters, denoted by $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$ that have to be determined during the training process. In this paper, we define $\mathcal{IS}^{\boldsymbol{\theta}}$ as the mapping from the input space to the latent space $\mathcal{Z}_l$:

$$\begin{aligned} \mathcal{IS}^{\boldsymbol{\theta}} : U &\to Z_l \\ \mathbf{u} &\mapsto \mathbf{z}_l, \end{aligned} \tag{8}$$

and denote by $\hat{\mathbf{z}}_l$ a prediction of the latent variables by the intermediate surrogate, *e.g.*, a Gaussian process. The dependence of $\mathcal{IS}^{\boldsymbol{\theta}}$ on $\boldsymbol{\theta}$ is omitted hereafter for the sake of conciseness.

There exist mainly two strategies to build a surrogate of a mapping with vector outputs: either build a set of independent surrogates with scalar outputs (one for each component of the output vector), or a single surrogate capable of predicting vector outputs, possibly using the dependence information between the output components.

The first strategy leads to $d_{\mathbf{z}_l}$ surrogates that have to be trained independently. They can be single-fidelity [39] or multi-fidelity [40, 41]. Gaussian process regression [42], polynomial chaos expansion [43], artificial neural networks [44], random forests [45] or radial basis function (RBF) interpolation [46] are well-known techniques to build such a single-fidelity surrogate. Multi-fidelity extensions include GP-based methods (*e.g.*, hierarchical Kriging [47], autoregressive (AR1) co-Kriging [48], nonlinear autoregressive co-Kriging [49], multi-fidelity deep Gaussian processes (DGP) [50, 51]), multi-fidelity polynomial chaos expansion [52], artificial neural networks [13] and co-RBF [53]. Such multi-fidelity techniques mainly differ from the dependence hypothesis between the different fidelity models (*e.g.*, linear, nonlinear).

The second strategy involves a single multi-output surrogate that can either be single-fidelity or multi-fidelity. Single-fidelity examples include multi-output GP regression [54], artificial neural networks[44], RBF interpolation [46]. Similarly, an important modeling characteristic lies in the dependence hypothesis between the components of the output (*e.g.*, linear, nonlinear).

In the literature, there are very few multi-fidelity surrogates capable of predicting multi-outputs. Among the existing ones, there are multi-outputs AR1 co-Kriging [12, 55], RBF interpolation with the addition of an input parameter to indicate the fidelity [11] or multi-fidelity DGP [50, 51].

## 3. Multi-fidelity surrogates for the prediction of high-dimensional field outputs

Multiple methods for building multi-fidelity surrogates combining DR and intermediate surrogate modeling for the prediction of field outputs have been proposed in the literature. However, these approaches have received little comparison and have not been presented within a common framework, making the comparison of their pros and cons difficult for the newcomer. They all share common building blocks, which have been introduced in Section 2, and their combination can be divided into three families of approaches: corrective, mapping and fusion. In the original papers [1–12], these surrogates have all been introduced for the case when two fidelities are available (*i.e.*, $l \in \{1, 2\}$). Therefore, the different families are subsequently introduced for the two-fidelity case, even though some could theoretically be extended to a greater number of fidelities. Additionally, the prediction process is only presented for the prediction of higher-fidelity simulator outputs. In this section, a unified framework is proposed that conveys the different paradigms in a pedagogical manner, facilitating the comparison of the different surrogates and their adaptation by practitioners.

### *3.1. Corrective approaches*

In the so-called *corrective approaches*, the high-fidelity output field is not modeled directly. The basic idea is to build a surrogate serving as a corrective function added to the low-fidelity simulator to predict high-fidelity output fields. In practice, the first step of the corrective approach is to compute the difference between the high- and low-fidelity snapshots:

$$\mathcal{Y}_\Delta = \left\{ \mathbf{y}_\Delta^{(i)} = \mathbf{y}_1^{(i)} - \mathbf{y}_2^{(i)} \right\}_{i=1}^{n} . \tag{9}$$

Note that in order to be able to compute this, there must be the same number $n = n_1 = n_2$ of high- and low-fidelity snapshots, corresponding to the same input variable vectors (*i.e.*, $\mathcal{U}_1 = \mathcal{U}_2 = \mathcal{U}$). Consequently, if the computational cost of the high-fidelity simulator is especially high, the

small number of high-fidelity snapshots will limit the number of exploitable low-fidelity snapshots. Hence, the surrogate may not take full advantage of the multi-fidelity context. Furthermore, either the high- and low-fidelity output fields must be discretized on the same mesh, or a pre-processing step transferring the fields onto a common mesh must be performed. Then, DR is carried out using the snapshots in $\mathcal{Y}_\Delta$ to determine the associated latent variables values

$$\mathcal{Z}_\Delta = \left\{ \mathbf{z}_\Delta^{(i)} = \mathcal{DR}\left(\mathbf{y}_\Delta^{(i)}\right) \right\}_{i=1}^{n}. \tag{10}$$

Finally, a single-fidelity surrogate $\mathcal{IS}_{\mathrm{SF}}$ is built to map the input space $U$ to the latent space using the samples from $\mathcal{U}$ and $\mathcal{Z}_\Delta$. It is then able to predict the latent variables for a given input variable vector. A diagram summarizing this approach is displayed in Figure 2. Any single-fidelity DR technique and single-fidelity surrogate modeling technique may be used to construct such a corrective surrogate.

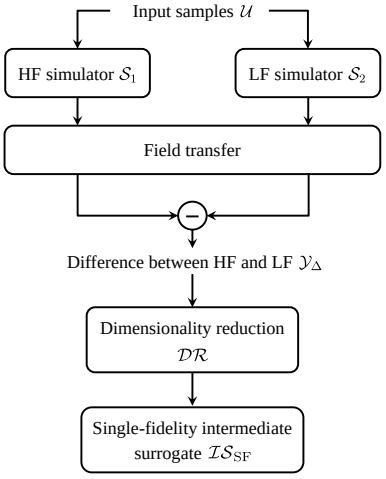



Figure 2: Diagram of the corrective approaches [4]

To perform a prediction for a new input vector $\mathbf{u}^\star \notin \mathcal{U}$, the surrogate is used to predict the latent variables $\hat{\mathbf{z}}_\Delta = \mathcal{IS}_{\mathrm{SF}}(\mathbf{u}^\star)$. Then, the DR is mapped back to get the prediction of the difference field $\hat{\mathbf{y}}_\Delta = \mathcal{DR}^{-1}(\hat{\mathbf{z}}_\Delta)$. Finally, $\hat{\mathbf{y}}_\Delta$ is added to the low-fidelity output field obtained by running the exact low-fidelity simulator $\mathcal{S}_2(\mathbf{u}^\star)$. The process of getting a high-fidelity output field prediction $\hat{\mathbf{y}}_1$ can be summarized by the following equation:

$$\hat{\mathbf{y}}_1 = \mathcal{S}_2\left(\mathbf{u}^\star\right) + \mathcal{DR}^{-1}\left(\mathcal{IS}_{\mathrm{SF}}(\mathbf{u}^\star)\right). \tag{11}$$

The multi-fidelity surrogate proposed by Malouin *et al.* [4] belongs to the corrective approaches. For the DR, the authors used PCA and the intermediate surrogate is a single-fidelity GP per latent variable.

### *3.2. Mapping approaches*

*Mapping approaches* consist in building a multi-fidelity surrogate that maps the output field of the low-fidelity simulator to the high-fidelity output field. The first step is to compute the sets of latent variable values corresponding of the high- and low-fidelity snapshots separately, respectively $\mathcal{Z}_1$ and $\mathcal{Z}_2$. They may be computed with different DR techniques, and are such that

$$\mathcal{Z}_l = \left\{\mathbf{z}_l^{(i)} = \mathcal{DR}_l\left(\mathbf{y}_l^{(i)}\right)\right\}_{i=1}^{n}, l \in \{1, 2\}. \tag{12}$$

The second step is to train a single-fidelity surrogate mapping the manifold $Z_2$, represented by the latent samples $\mathcal{Z}_2$, to the manifold $Z_1$, represented by the latent samples $\mathcal{Z}_1$. This surrogate can then predict the high-fidelity latent variables if the low-fidelity latent variables are provided. As for the corrective approach, the same number $n$ of high- and low-fidelity latent variables snapshots (corresponding to the same input variable vectors) is required to be able to define this mapping. Again, if acquiring high-fidelity simulator snapshots is computationally expensive, this may limit the number of low-fidelity snapshots that can be exploited. Hence, the surrogate may not take full advantage of the multi-fidelity context. Note that this mapping resembles manifold alignment as it maps the low- to the high-fidelity manifold. The difference is that it is used in the prediction process while manifold alignment is only used in the training process. The different steps involved in mapping surrogates are summarized in Figure 3. Different approaches for performing DR [9, 10] are displayed and represented by different colors (red for common DR [9] and yellow for separate DR [10]).

To predict the high-fidelity output field for a new input variable vector $\mathbf{u}^\star \notin \mathcal{U}$, it is necessary to first run the low-fidelity simulator to get the corresponding low-fidelity output field $\mathbf{y}_2^\star = \mathcal{S}_2(\mathbf{u}^\star)$. Then, the low-fidelity latent variables corresponding to this field are computed with $\mathbf{z}_2^\star = \mathcal{DR}_2(\mathbf{y}_2^\star)$. Next, the single-fidelity surrogate is used to predict the high-fidelity latent variables $\hat{\mathbf{z}}_1 = \mathcal{IS}_{\mathrm{SF}}(\mathbf{z}_2^\star)$. Finally, the inverse mapping of the high-fidelity

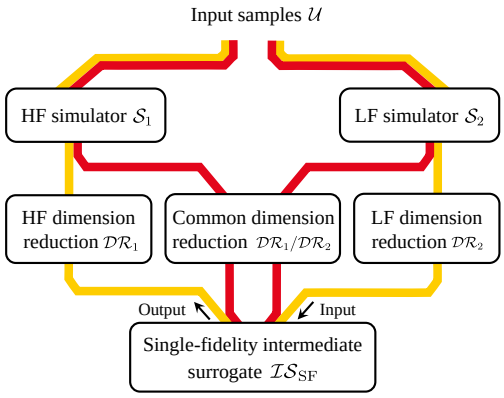



Figure 3: Diagram of the mapping approach, [9] in red, [10] in yellow. The arrows represent the input and output of the single-fidelity intermediate surrogate $\mathcal{IS}$.

DR is performed to get a prediction $\hat{\mathbf{y}}_1$ of the high-fidelity output field. This process is summarized in Equation (13).

$$\hat{\mathbf{y}}_1 = \mathcal{DR}_1^{-1}\left(\mathcal{IS}_{\mathrm{SF}}(\mathcal{DR}_2(\mathcal{S}_2(\mathbf{u}^\star)))\right). \tag{13}$$

The multi-fidelity surrogate proposed by Wang *et al.* [10] belongs to the mapping approach family (yellow path in Figure 3). The DR is performed using PCA independently for each fidelity level, hereafter referred to as single-fidelity PCA (SFPCA). The mapping of the latent variables between the fidelity levels is a Gaussian process per high-fidelity latent variable. Note that if the PCA of the low-fidelity snapshots results in a large number of latent variables, this mapping has a large input dimensionality, which might make the learning process more complex. Additionally, classical GP kernels use Euclidean distances in the input space to measure the distance between the input samples. If nonlinear DR has been selected, the input space is not an Euclidean space. Therefore, the kernel should be adapted for instance by being based on geodesic distances instead (*e.g.*, see [56]). In [10], a GP with a kernel based on Euclidean distances is used. We make the same choice in the sequel of this paper.

Another multi-fidelity surrogate in the mapping family has been proposed by Toal [9] (red path in Figure 3). It uses Gappy-PCA (GPCA, most commonly known as Gappy-POD) [57] which performs a PCA over the stacked high- and low-fidelity snapshots, *i.e.*, over $\mathbf{Y}_{\mathrm{GPCA}} = [\mathbf{Y}_1, \mathbf{Y}_2]$, which is a $n$-by-$(d_{\mathbf{y}_1} + d_{\mathbf{y}_2})$ matrix. As used by Toal [9], GPCA is not only a DR technique,

but also defines a mapping from $Y_2$ to $Y_1$. The low-fidelity simulator has to be run in order to perform a prediction of the high-fidelity output field.

### *3.3. Fusion approaches*

*Fusion approaches* combine DR and an intermediate surrogate model blending together the high- and low-fidelity data in such a way that running the exact low-fidelity simulator is not needed to predict high-fidelity output fields. The first step, which is performing DR, is the same as mapping approaches. The latent samples corresponding to the high- and low-fidelity snapshots sets, respectively $\mathcal{Z}_1$ and $\mathcal{Z}_2$, are computed with

$$\mathcal{Z}_l = \left\{ \mathbf{z}_l^{(i)} = \mathcal{DR}_l \left( \mathbf{y}_l^{(i)} \right) \right\}_{i=1}^{n_l}, l \in \{1, 2\}. \tag{14}$$

To perform this DR, there are two main options. The first uses the same mapping to the latent space for both fidelity levels. This causes the high- and low-fidelity latent variables to belong to the same manifold. The second option uses two distinct DR processes for each fidelity level, resulting in two distinct manifolds. For fusion approaches, it is possible to have a different number of snapshots for different fidelities (*i.e.*, $n_1 \neq n_2$). An optional step is to perform manifold alignment by orthogonal Procrustes analysis that transfers the known points of the low-fidelity manifold $Z_2$ to the high-fidelity manifold $Z_1$. After applying the operations described in Section 2.2, the low-fidelity latent variable values $\mathcal{Z}_2$ would become $\mathcal{Z}_{2'}$ ($\mathbf{z}_{2'} = \mathcal{MA}(\mathbf{z}_2)$). Eventually, a multi-fidelity intermediate surrogate $\mathcal{IS}_{\text{MF}}$ is built that maps the input space $U$ to the high-fidelity latent space $Z_1$. It mixes the high- and low-fidelity training data $\mathcal{U}_1$, $\mathcal{Z}_1$, $\mathcal{U}_2$ and $\mathcal{Z}_2$ (or $\mathcal{Z}_{2'}$ if manifold alignment has been performed).

No matter the options chosen so far, the high- and low-fidelity latent spaces $Z_1$ and $Z_2$ must have the same dimensionality $d_{\mathbf{z}_1} = d_{\mathbf{z}_2} = d_{\mathbf{z}}$. This must be taken into account when performing DR. For instance, if DR is performed separately for each fidelity (resulting in manifolds having different dimensionalities), the common dimensionality could be the lowest of all manifold dimensionalities. This would lead to suboptimal DR for manifolds that would have had a higher dimensionality otherwise. An additional optional step is to compute the DR residuals of the high-fidelity output field and to train a single-fidelity surrogate $\mathcal{IS}_{\text{SF}}^{\perp}$ for predicting these residuals (see Section 2.3). Fusion methods are summarized in Figure 4.

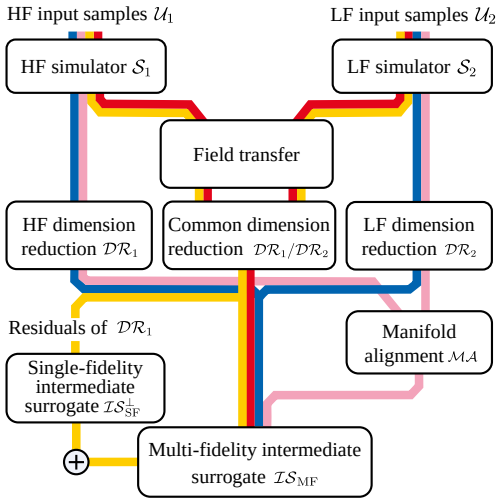



Figure 4: Diagram of the fusion approaches, [1, 7, 8, 11] in red, [3] in yellow, [2] in blue, [5, 6] in pink

To predict the high-fidelity output corresponding to a new vector of input variables $\mathbf{u}^\star \notin \mathcal{U}_1$, the multi-fidelity intermediate surrogate is used to predict the high-fidelity latent variables $\hat{\mathbf{z}}_1 = \mathcal{IS}_{\mathrm{MF}}(\mathbf{u}^\star)$. Then, the prediction of the high-fidelity output field $\hat{\mathbf{y}}_1$ is computed by inverse mapping the high-fidelity DR $\hat{\mathbf{y}}_1 = \mathcal{DR}_1^{-1}(\hat{\mathbf{z}}_1)$. Finally, if needed, the surrogate $\mathcal{IS}_{\mathrm{SF}}^{\perp}$ can be used to predict the residuals not taken into account in the DR. Note that manifold alignment does not imply any additional step in the prediction process, as it only modifies the training set of the multi-fidelity intermediate surrogate $\mathcal{IS}_{\mathrm{MF}}$ by replacing $\mathcal{Z}_2$ with $\mathcal{Z}_{2'}$. The following equation summarizes the prediction process of fusion surrogate models.

$$\hat{\mathbf{y}}_1 = \mathcal{DR}_1^{-1}\left(\mathcal{IS}_{\mathrm{MF}}\left(\mathbf{u}^\star\right)\right) + \mathcal{IS}_{\mathrm{SF}}^{\perp}(\mathbf{u}^\star). \tag{15}$$

In the following paragraphs, different multi-fidelity surrogates introduced in the literature belonging to the fusion family are briefly described. First, the surrogates that perform separate DR for each fidelity are presented. These surrogates have the advantage that the high- and low-fidelity output fields can be discretized on different meshes. The surrogate introduced by Bunnell *et al.* [2] (blue path in Figure 4), uses SFPCA and one AR1 co-Kriging per latent variable as an intermediate surrogate. This assumes that without any adjustment, the high- and low-fidelity manifolds are similar even though the

DRs are performed separately, since AR1 co-Kriging assumes a linear dependency between the high- and low-fidelity latent variables. In their paper, Bunnell *et al.* verified this similarity by plotting and visually comparing the PCA modes. This also assumes that there is the same number of high- and low-fidelity latent variables. This can either be done by *a priori* setting a fixed number of modes for all fidelities, or by choosing the number of modes of one of the fidelities as the dimensionality of all latent spaces.

The multi-fidelity surrogates proposed in Perron *et al.* [5] and extended in Decker *et al.* [6] also separate high- and low-fidelity DR (pink path in Figure 4). The used DR techniques are PCA [5], KPCA, Isomap and LTSA [6] which are performed on each fidelity independently, hence called SFPCA, SFKPCA, SFIsomap and SFLTSA. Manifold alignment by orthogonal Procrustes analysis is performed before using one hierarchical Kriging [47] per latent variable as a multi-fidelity intermediate surrogate. These surrogates are the only ones using manifold alignment. In their study, Perron *et al.* [5] and Decker *et al.* [6] considered the same number of high- and low-fidelity latent variables (this assumption could be removed, see Section 2.2 for more details).

Other multi-fidelity surrogates of the fusion family use a common DR for all fidelity levels. Consequently, they all share the same latent manifold. Note that the high- and low-fidelity output fields need to be discretized on the same mesh. The multi-fidelity surrogate introduced by Benamara *et al.* [1] (red path in Figure 4) uses an adaptation of Constrained PCA (CPCA). It consists in an untruncated PCA of the high-fidelity snapshots to which are added vectors computed from the part of the low-fidelity snapshots which is orthogonal to the basis. Then, intermediate surrogate modeling is performed with one AR1 co-Kriging per latent variable.

The multi-fidelity surrogate in Mifsud *et al.* [11] performs PCA over the union of the high- and low-fidelity snapshots $\mathcal{Y}_1 \cup \mathcal{Y}_2$ denoted here by MFPCA (red path in Figure 4). Since the number of snapshots is $n_1 + n_2$ and because PCA decomposes the variance of both the high- and low-fidelity output fields, the number of principal components will likely be larger than for a PCA of the high- or low-fidelity snapshots carried out seperately. Moreover, if the number of low-fidelity snapshots $n_2$ is much larger than the number of high-fidelity snapshots $n_1$, there is a risk of diluting the high-fidelity output field features when performing MFPCA. The intermediate mapping is a single-fidelity RBF with an additional discrete variable that indicates the fidelity of the corresponding snapshot (*i.e.*, $\mathbf{u}_l \in U \subseteq \mathbb{R}^{d_\mathbf{u}}$ is replaced by $\tilde{\mathbf{u}}_l \in \tilde{U} \subseteq$

$\mathbb{R}^{d_{\mathbf{u}}} \times \mathbb{N}$).

The surrogate described in Thenon *et al.* [8] uses a PCA over the high-fidelity snapshots (HFPCA) and one AR1 co-Kriging per latent variable as the intermediate mapping (red path in Figure 4). This assumes that the modes of HFPCA can accurately compress the low-fidelity output field. This also presumes that the number of high-fidelity snapshots is large enough to build a meaningful covariance matrix to compute its eigenvectors and eigenvalues.

The surrogate introduced by Rokita *et al.* [7] performs MFPCA and uses one AR1 co-Kriging per latent variable as the intermediate mapping (red path in Figure 4. The surrogate of Kerleguer [3] uses two different DR techniques (yellow path in Figure 4). The first one is a PCA on the low-fidelity snapshots (denoted here by LFPCA) and the other is a cross-validation based PCA over the low-fidelity snapshots (low-fidelity cross-validation based PCA, LFCVBPCA). An empirical distribution of the modes is obtained by considering PCAs of subsets of all snapshots as realizations of the CVBPCA. The intermediate surrogate is one AR1 co-Kriging per latent variable and the residuals of the high-fidelity DR is modeled by a Gaussian process with tensorized covariance. Note that this surrogate is the only one to model the residuals of DR. This surrogate either assumes that the modes computed from the low-fidelity snapshots (with LFPCA or LFCVBPCA) can accurately represent the high-fidelity output field, or that the surrogate of the residuals of DR can compensate the difference.

### *3.4. General remarks on the different approaches*

The main drawback of mapping and corrective approaches compared to fusion approaches is that they require the evaluation of the low-fidelity simulator to make a prediction of the high-fidelity output field. If the low-fidelity simulator has a non-negligible cost, this can make the use of these surrogates intractable in an optimization or uncertainty quantification context. To overcome this issue, the lower-fidelity simulator $\mathcal{S}_2$ could be replaced by a single-fidelity surrogate (for instance, a combination of PCA and GP regression). In Perron *et al.* [5], the surrogate using the corrective approach and the GPCA-based surrogate are compared when the low-fidelity simulator is used to make predictions of the high-fidelity output field and when it is replaced by a single-fidelity surrogate. In the examples considered in their study, replacing the low-fidelity simulator by an emulator does not always prove to be detrimental. Hence, it is hard to predict if this procedure should

improve or worsen the performance of multi-fidelity surrogate models in the general case.

Another disadvantage of mapping and corrective methods is the fact that the same amount of snapshots for the same input samples is required for each fidelity. Usually, since the low-fidelity simulator has a lower computational cost than the high-fidelity simulator, more low-fidelity snapshots are available. Mapping and corrective methods are not able to take advantage of this situation. This might not be the case if the low-fidelity simulator is replaced by a single-fidelity surrogate, as the latter could be trained with the additional low-fidelity snapshots.

Another point that is of great importance is the correspondence between the different fidelity meshes. The corrective and fusion approaches having a common DR for both fidelity levels (*i.e.*, MFPCA, HFPCA and LFCVBPCA), they require that meshes are the same for each fidelity level. If this is not the case, a pre-processing step projecting all snapshots to a common mesh must be carried out. This common mesh can be the high-fidelity mesh, the low-fidelity mesh or neither of those. If the chosen common mesh is not the high-fidelity mesh, the prediction might have to be mapped from the common mesh to the high-fidelity mesh.

The multi-fidelity surrogate of Benamara *et al.* [1] has nested meshes for the different fidelity levels. The chosen common mesh is the low-fidelity mesh, hence, only the common nodes of the high- and low-fidelity meshes are used to build the multi-fidelity surrogate. In Malouin *et al.* [4], both fidelity levels are projected onto the high-fidelity mesh with a "nearest neighbor" strategy. In Thenon *et al.* [8], the common mesh is also the high-fidelity mesh. The interpolation method used is specific to reservoir engineering [58]. In Rokita *et al.* [7], both fidelity levels are projected onto the low-fidelity mesh. In Mifsud *et al.* [11], the high-fidelity mesh is chosen as the common mesh. In the work of Kerleguer [3], it is assumed that both fidelity levels are discretized on the same mesh. Conversely, mapping approaches do not require the mesh to be the same for all fidelity levels, except if a common DR is used.

The total number of hyperparameters varies between the various multi-fidelity surrogates presented so far. First, there are hyperparameters that have to be set prior to training the surrogate. They can have a strong impact on the performance of both DR and intermediate surrogate modeling. All DR techniques reviewed in this paper have such hyperparameters: for instance, the RIC in the truncation of PCA, the dimensionality of the latent space in Isomap and LTSA, and the kernel function of KPCA all have to be chosen

beforehand, based on experience of the practitioner or any specific technique depending on the method. For the mapping from the input to the latent space, some hyperparameters of the selected surrogate must be set *a priori*, *e.g.*, the kernel function when using Gaussian processes.

Second, some hyperparameters are tuned when the surrogate is trained. Note that they could be set beforehand instead. For instance, in DR, the number of nearest neighbors in Isomap and LTSA can be chosen so as to minimize a metric (Kruskal's stress [59] for Isomap and the variance of distance ratios [60] for LTSA). Similarly, the correlation length of the kernel is optimized in KPCA. In general, the more hyperparameters there are, the more complex the computation of the DR mapping. This is one of the advantage of PCA compared to nonlinear DR techniques. Considering the intermediate surrogate, the correlation length of the kernel function of Gaussian processes is typically determined by maximizing the likelihood [61].

The above approaches could, at least theoretically, be extended to more than two fidelity levels ($l \in \{1, \dots, L\}$) in rather straightforward manners. In practice, the increase in computational cost may become a limiting factor. For corrective approaches, the following extension could be used:

$$\hat{\mathbf{y}}_1 = \mathcal{S}_L(\mathbf{u}^\star) + \sum_{l=1}^{L-1} \Delta\hat{\mathbf{y}}_{l,l+1}, \tag{16}$$

with $\Delta\hat{\mathbf{y}}_{l,l+1}$ a single-fidelity surrogate of the difference between the simulators of fidelity $l$ and $l+1$, evaluated at $\mathbf{u}^\star$. The lowest-fidelity simulator could be used if its computational cost is sufficiently low or it could be replaced by a single-fidelity surrogate otherwise. Mapping approaches could be extended by using the latent variables of all fidelity levels except the highest one as the inputs of the surrogate predicting the high-fidelity latent variables, *i.e.*, $\hat{\mathbf{z}}_1 = \mathcal{IS}_{\text{SF}}\left(\mathbf{z}_{[2,L]}\right)$, with $\mathbf{z}_{[2,L]} = \left[\mathbf{z}_2^\top, \dots, \mathbf{z}_L^\top\right]^\top$. Note that the dimensionality of the surrogate input $\mathbf{z}_{[2,L]}$ could become very large. If this extension is adopted, a special care should be given to the choice of the intermediate surrogate so that it supports high-dimensional inputs.

Fusion approaches require little modification to support more than two fidelity levels. Indeed, only the multi-fidelity surrogate might need to be changed to be able to use training data from more than two fidelity levels. For instance, AR1 co-Kriging or RBF interpolation with an additional input variable indicating the fidelity level could be used. As discussed in Section 2.2, manifold alignment can be performed with more than two fidelity

levels. Since the residuals of DR are only modeled for the highest-fidelity, this require no modification.

## 4. Description of the test cases

Before introducing the benchmark methodology, this section presents the different test cases that are used to evaluate the performance of the different multi-fidelity surrogates with functional outputs. The physics of the problems is described and implementation details are given.

### *4.1. Viscous free fall of a ball*

The first test case deals with the free fall of a ball thrown in a fluid medium. It is derived from [62]. The goal is to build a surrogate of the trajectory of the ball, which is a time series, as a function of the initial altitude $y_0 \in [0.2, 0.4]$ m, the initial vertical velocity $v_0 \in [10, 20]$ m.s$^{-1}$, the uniform density of the ball $\rho_{\text{ball}} \in [10, 100]$ kg.m$^{-3}$ and the radius of the ball $r \in [0.1, 1]$ m.

The ball has a mass $m = \rho_{\text{ball}} \times \frac{4\pi r^3}{3}$. The altitude of the ball at time $t \in [0, T]$ is denoted by $y(t)$. At the initial time, it is given a vertical velocity $\dot{y}(0) = v_0$ and an altitude $y(0) = y_0$. Then, it is only subjected to the acceleration of gravity $g = 9.81$ m.s$^{-2}$ and the viscous drag force $F_d = \frac{1}{2} C_d(\dot{y}(t))\, \rho_{\text{fluid}}\, a\, \dot{y}(t)^2$ due to the resistance of the fluid it is immersed in, with the drag coefficient $C_d(\dot{y}(t))$ which depends on the velocity $\dot{y}(t)$, the density of the fluid $\rho_{\text{fluid}}$ and the cross-sectional area $a$ ($\pi r^2$ for a sphere of radius $r$). The fluid is water at 20°C with a density $\rho_{\text{fluid}} = 998.3$ kg.m$^{-3}$ and a dynamic viscosity is $\eta = 1.002 \times 10^{-3}$ Pa.s. The governing movement equation is given by

$$m\ddot{y}(t) + F_d(\dot{y}(t)) + mg = 0. \tag{17}$$

The input variables of the simulators are represented by the vector $\mathbf{u} = [y_0, v_0, \rho_{\text{ball}}, r]^\top$. The simulators of variable fidelity all solve Equation (17) for a finite number $d_{\mathbf{y}}$ of time steps evenly spaced on $[0, T]$, *i.e.*, the mesh is of size $d_{\mathbf{y}}$. The altitude at each time step is collected in the vector $\mathbf{y}_l$, which forms the output field of the simulator of fidelity $l$. The difference between the fidelities stems from the modeling of the drag coefficient.

In the low-fidelity simulator, the drag coefficient is supposed to be constant. It is chosen to be equal to 0.4 [63]. It can therefore be shown that

the drag force is $F_d(\dot{y}(t)) = 0.2\,\pi\,\rho\,r^2\,\dot{y}(t)^2$. Equation (17) can be solved analytically, which gives the solution

$$y_2(t) = y_0 + v_{inf}t + \tau(v_0 - v_{inf})\left(1 - e^{-\frac{t}{\tau}}\right), \tag{18}$$

with $\tau = \frac{m}{c}$ the characteristic time of the system, $v_{inf} = \frac{-mg}{c}$ the terminal velocity and $c = \frac{1}{2}C_d(\dot{y}(t))\,\rho_{\text{fluid}}\,a$. The output field of the low-fidelity simulator is $\mathbf{y}_2 = [y_2(0), y_2(T/d_{\mathbf{y}}), \ldots, y_2(T)]^\top$.

In the high-fidelity simulator, the drag coefficient is supposed to be the following function of the velocity [63]

$$C_d(\dot{y}(t)) = \frac{24}{Re} + \frac{6}{1+\sqrt{Re}} + 0.4, \tag{19}$$

with $Re = \frac{\rho_{\text{fluid}}\ell\dot{y}(t)}{\eta}$ the Reynolds number, $\eta$ the dynamic viscosity of the medium and $\ell$ the characteristic length scale of the ball in the cross-sectional plane (which is here the diameter $2r$). With this drag model, Equation (17) cannot be solved analytically. Consequently, it is solved numerically with the LSODE algorithm [64] and the solution is extracted on the mesh nodes. The output field of the high-fidelity simulator is $\mathbf{y}_1 = [y_1(0), y_1(T/d_{\mathbf{y}}), \ldots, y_1(T)]^\top$.

From both fidelities, two related test cases are derived. First, the trajectory is considered in the absence of a ground. This configuration does not imply any modification to the problem introduced above. Samples of this configuration are shown in Figure 6. Second, a ground is added at altitude 0 m without rebound on impact, thus introducing a discontinuity when the sphere touches the ground. This is performed by replacing every negative value of the altitude by a zero. Samples of this configuration are depicted in Figure 7.

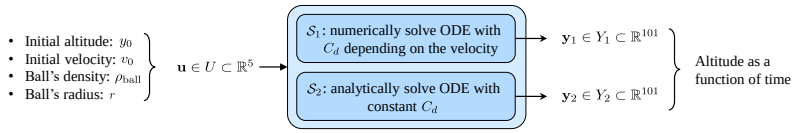



Figure 5: Simulation setup for the free fall of a ball in a fluid medium case.

It can be seen that the shape of the high- and low-fidelity trajectories are similar. For the high-fidelity, the amplitude is higher and the terminal velocity is reached earlier. The impact with the ground is reached earlier in the low-fidelity simulations. Note that the simulation duration is chosen

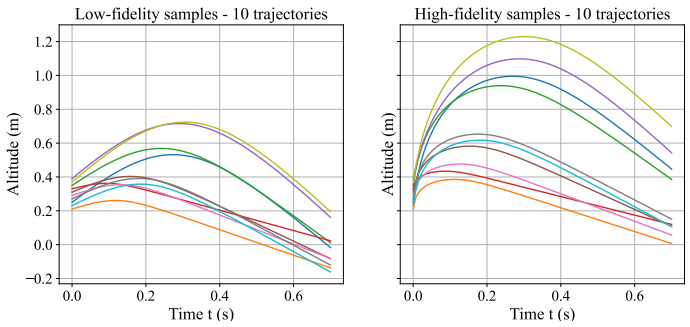



Figure 6: 10 samples of the high- and low-fidelity output fields for the viscous free fall of a ball in a fluid in the absence of a ground. High- and low-fidelity fields corresponding to the same input samples are in the same color.

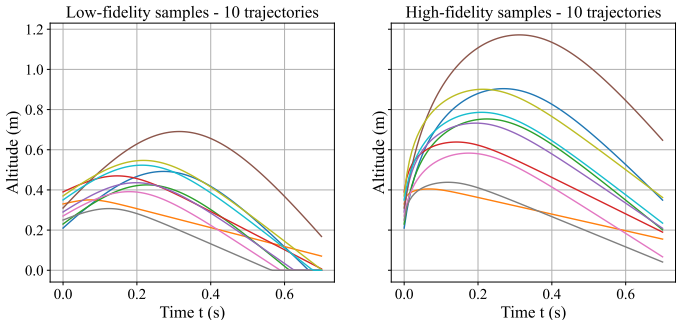



Figure 7: 10 samples of the high- and low-fidelity output fields for the viscous free fall of a sphere in a fluid in the presence of a ground. High- and low-fidelity fields corresponding to the same input samples are in the same color.

such that some low-fidelity simulations reach the ground while some do not. The time of impact is also varying among the different samples.

### *4.2. One-dimensional pressure coefficient field at the wall of a NACA 0015 airfoil*

The second test case is the study of the aerodynamic properties of an airfoil in a subsonic flying regime. The quantity of interest is the pressure coefficient $c_p$ field at the upper surface of the airfoil, which is defined as $(p - p_\infty)/q_\infty$, with $p$ the static pressure at the point of evaluation, $p_\infty$ the freestream static pressure and $q_\infty$ the freestream dynamic pressure. The goal is to build a surrogate of this spatial $c_p$ field as a function of the angle of

attack and of the deformation of the airfoil geometry.

The baseline airfoil is a symmetrical NACA 0015 with a chord length of 1 m. The geometry is parameterized by 35 control points spread out along the surface. They are numbered starting from the trailing edge, then going along the upper surface from right to left, then along the lower surface from left to right, finishing by the trailing edge again. The control points n°5, 14, 21 and 33 are allowed to move vertically as can be seen in Figure 8 in order to alter the shape of the airfoil. The shape of the wing is then interpolated from these control points with Bernstein polynomials. The angle of attack $\alpha$ varies between 4° and 6°.

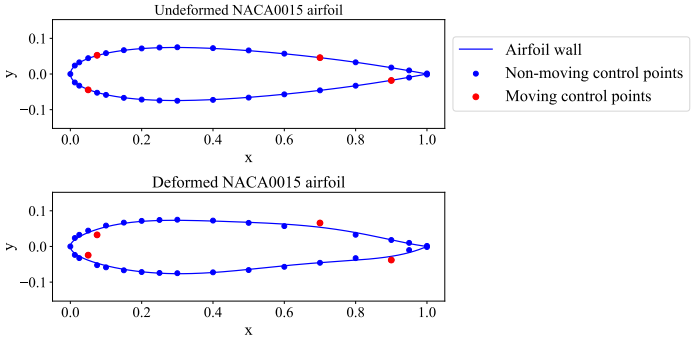



Figure 8: Undeformed airfoil (top) and deformed airfoil (bottom). The deformation of the airfoil is amplified for illustration purposes.

The input variables of the simulators are represented by the vector $\mathbf{u} = (\alpha, \delta_5, \delta_{14}, \delta_{21}, \delta_{33})^\top$, with $\delta_i \in [-0.01, 0.01]$ m the vertical displacement of the $i$-th control point from its nominal position. The simulations are performed using the python package `aeropy` [65] which itself uses XFOIL [66] to solve the CFD problem with a panel method. For both the high- and low-fidelity simulators, the upper and lower surfaces of the airfoil are discretized into 80 nodes each. After running the simulators, the pressure coefficient $c_p$ is known at the 160 nodes. The 80 $c_p$ values on the upper surface are then interpolated on a finer mesh of 1001 nodes evenly spaced along the chord line. The interpolator is the piecewise cubic Hermite interpolating polynomial of `scipy` [67]. For both the high- and low-fidelity, the output of the simulator is the $c_p$ field at the upper surface of the airfoil on this finer mesh. In the low-fidelity simulator, the fluid is supposed to be inviscid. In the high-fidelity simulator, the viscosity is taken into account, and the Reynolds number is

$10^6$. The simulation setup is summarized in Figure 9 and samples of both the high- and low-fidelity simulators are depicted in Figure 10.

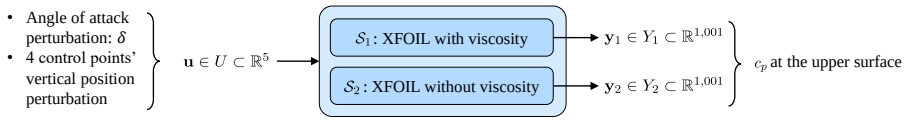



Figure 9: Simulation setup for the NACA 0015 airfoil case

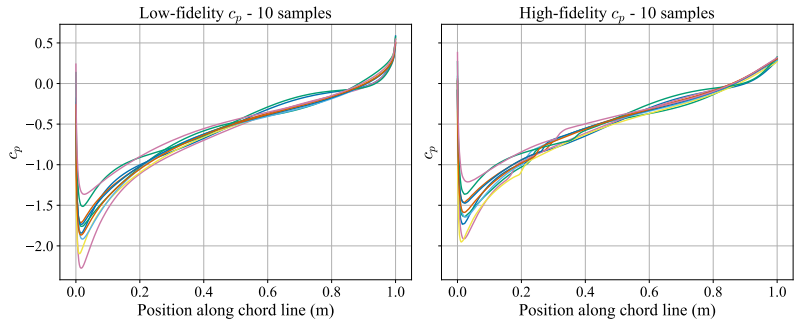



Figure 10: Ten samples of the high- and low-fidelity output fields for the NACA 0015 baseline airfoil case. High- and low-fidelity fields corresponding to the same input samples are in the same color.

It can be seen that the high- and low-fidelity simulators output very similar $c_p$ fields. The most notable difference is the "step" around 0.25 m along the chord line for the high-fidelity samples. This step corresponds to the boundary layer separation. Notice that this step is not located at the same spatial position between snapshots.

### *4.3. Two-dimensional pressure coefficient field around an airfoil*

The last test case is the study of the aerodynamics of a transonic airfoil in a subsonic regime. This test case comes from a study of Perron *et al.* [5] from which the data has been made available by the authors. The aim is to build a surrogate of the pressure coefficient $c_p$ in the 2D flow around the airfoil as a function of the angle of attack and the deformation of the airfoil.

The baseline airfoil is as RAE 2822 which is immersed in an airflow with a freestream Mach number of 0.725. The angle of attack is varied between

0° and 4°. The airfoil is deformed by Free Form Deformation (FFD) [68]. It is parameterized by two FFD boxes that can only move vertically, with a displacement ranging from -0.03 to 0.03 chord length. The quantity of interest is the pressure coefficient field in the 2D spatial flow field. This simulation can typically be used to study the aerodynamic properties of the airfoil.

The datasets provided by Perron *et al.* consists of the simulation of this physical model for three different levels of fidelity. For each fidelity, the CFD calculation are carried out using the Stanford University Unstructured (SU2) code suite [69]. Additionally, the mesh at each fidelity is a structured grid, yet with varying resolution. When the surface of the airfoil is deformed by FFD, the deformation is propagated to the mesh using a linear elasticity approach [70]. This method ensures that for a given fidelity, the number of nodes and connectivity remain the same for every snapshot. Note that it also implies that the nodes have varying coordinates between the snapshots of a given fidelity.

The simulator of higher-fidelity is a CFD RANS simulator with a Spalart-Allmaras turbulence model [71]. The finest mesh comprises 41,796 nodes. The intermediate fidelity simulator is also a CFD RANS simulator with a Spalart-Allmaras turbulence model but with a coarser mesh of 10,530 nodes. The lower-fidelity simulator is a CFD Euler simulator with an even coarser mesh of 8,910 nodes with no near-wall refinement. For more details about the implementation, the reader is referred to the work of Perron *et al.* [5].

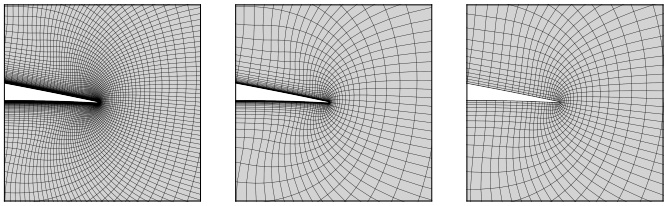


Figure 11: Zoom on the trailing edge of mesh $\mathbf{X}_1$ (left), $\mathbf{X}_2$ (center), $\mathbf{X}_3$ (right)

Since not all of the reviewed multi-fidelity surrogates support more than two fidelities, the benchmark is limited to the case where only two fidelities are available. Two possible configurations are considered in this paper. First, the RANS/RANS configuration is the combination of the CFD RANS with a fine mesh as the high-fidelity simulator $\mathcal{S}_1$ and the CFD RANS with a coarser

mesh as the low-fidelity simulator $\mathcal{S}_2$. This configuration yields a high correlation between the high- and low-fidelity output fields since the physical hypotheses of CFD RANS is the same for both fidelities. Second, the combination of the CFD RANS with a fine mesh as the high-fidelity simulator $\mathcal{S}_1$ and the CFD Euler with a coarser mesh as the low-fidelity simulator $\mathcal{S}_2$ is referred to as the RANS/Euler configuration hereinafter. This second configuration has a lower correlation between the high- and low-fidelity output fields since the physical hypothesis of CFD RANS and CFD Euler are not the same (no viscosity model for CFD Euler).

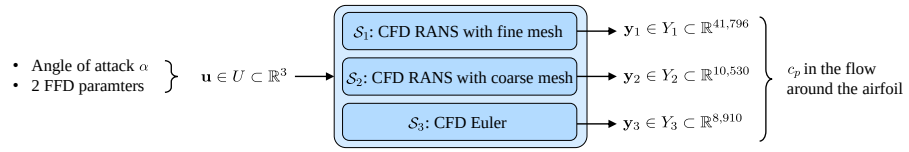



Figure 12: Simulation setup for the transonic airfoil case

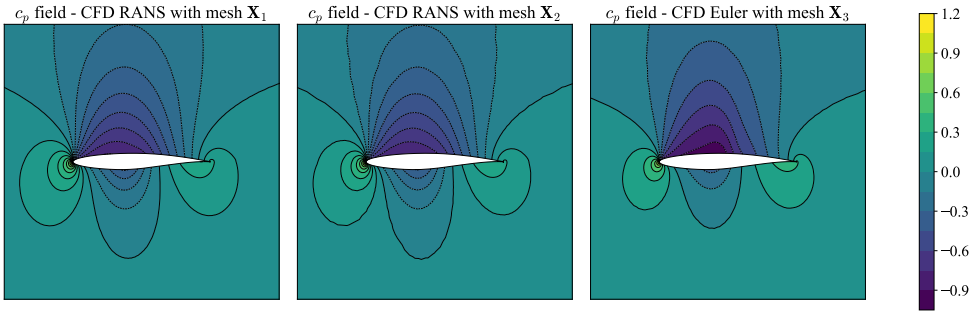



Figure 13: One sample of the CFD RANS with a fine mesh (left), CFD RANS with a coarser mesh (center) and CFD Euler (right) fields for the RAE 2822 airfoil without deformation and an angle of attack of 0°

## 5. Benchmark methodology

### *5.1. Training and validation data*

The generation of the training and validation data has been designed to be consistent across all the test cases. The number $n_1$ of high-fidelity training samples is proportional to the dimensionality of the input space $\dim(U)$, *i.e.*,

there are $n_1 = \{2, 5, 10\} \times \dim(U)$ high-fidelity training samples. Similarly, the number $n_2$ of low-fidelity training samples is chosen proportional to the number of high-fidelity samples. We choose $n_2 = \{1, 5, 10\} \times n_1$. This gives 9 combinations of $(n_1, n_2)$.

To assess the robustness of the multi-fidelity surrogates to the training DoE, ten repetitions are considered with different DoEs for each of these combinations. A validation set of $n_v = 1{,}000$ samples that are all different from the training samples is also created.

For a given repetition, the value of the input training samples is chosen with a nested Latin Hypercupe Sampling (LHS) strategy, as some multi-fidelity surrogates evaluated in the benchmark require a nested DoE. In this work, the LHS are created with a modified version of the `smt` [72] toolbox. It generates a first LHS of $n_1$ samples in $U$ that constitutes the high-fidelity input DoE $\mathcal{U}_1$. Then, a second LHS is built with $n_1 + n_2$ samples. From this second LHS, the sample closest in terms of Euclidean distance to the first sample of $\mathcal{U}_1$ is removed. This operation is repeated for every sample in $\mathcal{U}_1$. $\mathcal{U}_2$ is finally obtained by the union of the remaining samples with $\mathcal{U}_1$, hence the name nested. Note that this nested scheme implies that $\mathcal{U}_1 \subset \mathcal{U}_2$. The output DoE $\mathcal{Y}_1$ (respectively $\mathcal{Y}_2$) is obtained by running the high-fidelity (respectively low-fidelity) simulator over the samples in $\mathcal{U}_1$ (respectively $\mathcal{U}_2$). Surrogates requiring the same number of high- and low-fidelity snapshots (*i.e.*, corrective and mapping approaches) only use the common part of the design of experiments, that is only the high- and low-fidelity snapshots corresponding to the input samples in $\mathcal{U}_1$.

The specific case of the transonic airfoil requires an additional step since the samples are already available while generating new ones is not possible. For a given repetition, after the generation of the nested LHS, each input DoE sample is replaced by the closest input sample in terms of Euclidean distance from the Perron *et al.* dataset, making sure that no sample from the Perron *et al.* [5] dataset is used more than once. The output DoE samples are then replaced by the corresponding output samples from the Perron *et al.* [5] dataset.

To ensure a representative comparison of the surrogates, the intermediate surrogate models have been streamlined. More precisely, each single-fidelity intermediate surrogate has been replaced by ordinary Kriging [61] and each multi-fidelity Kriging by AR1 co-Kriging [48]. The practical implementation used for the benchmark and the naming of the surrogates is summarized in Table 1. The python code of the surrogate models as well as the training and

validation data are available upon request. All the multi-fidelity surrogates are compared to two reference single-fidelity surrogates, trained using only the high-fidelity snapshots. The first one is based on PCA with one Gaussian process per latent variable. The second one is a Gaussian process with covariance tensorization.

Some of the reviewed surrogates require that the low- and high-fidelity output fields are discretized on the same mesh. This is not a problem for the viscous free fall and the NACA 0015 airfoil since the low- and high-fidelity meshes are the same. For the RAE 2822 airfoil case, all fidelities have a different mesh. Since the surrogates that do not require a common mesh are able to predict high-fidelity output fields on the high-fidelity mesh, the high-fidelity mesh is chosen as the common mesh for surrogates that require the same mesh for each fidelity level. Whenever needed, the low-fidelity output field (either the CFD RANS with a coarse mesh or the CFD Euler) is transferred to the high-fidelity mesh with a nearest neighbor interpolation method (see Table 1). The surrogate models of Kerleguer *et al.* [3] have been implemented but due to the very large computational cost of the CVBPCA, the surrogate using the latter could not be included in the benchmark. The manifold alignment technique implemented here is the Procrustes analysis as explained in Section 2.2. More details about the implementation of the methods can be found in Appendix A.

### *5.2. Comparison metrics*

The performance of the different surrogates is estimated through their prediction accuracy. Predictions are made for the $n_v$ samples of the input validation set $\mathcal{U}_v$. The root-mean-square (rms) of the difference between the validation snapshots and the predictions is computed, which gives the rms error

$$e = \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \left|\left| \mathbf{y}_1^{(i)} - \hat{\mathbf{y}}_1^{(i)} \right|\right|^2}, \tag{20}$$

where $\mathbf{y}_1^{(i)} = \mathcal{S}_1 \left( \mathbf{u}_l^{(i)} \right)$ is the evaluation of the high-fidelity simulator and $\hat{\mathbf{y}}_1^{(i)}$ is the prediction of the tested surrogate for the $i$-th sample of $\mathcal{U}_v$.

To be comparable across different use cases, this error can be normalized by dividing $e$ by the rms of the difference between the validation snapshots and their mean value $\bar{\mathbf{y}}_1$, such that

Table 1: Summary of multi-fidelity surrogates for the prediction of field outputs (for the type of surrogate model: C=Corrective, M=Mapping, F=Fusion, TensCovGPR=Kriging with tensorized covariance, S=Single-fidelity; for dimensionality reduction: SF=Single-Fidelity, MF=Multi-fidelity, HF=High-Fidelity, LF=Low-Fidelity, PCA=Principal component Analysis, DiffPCA=PCA of the difference between the high- and low-fidelity snapshots, CPCA=Contrained PCA, GPCA=Gappy-PCA, KPCA=Kernel PCA, LTSA=Local Tangent Space Alignment, MA=Manifold Alignment; and for the intermediate surrogate: GPR=Gaussian process regression, CategGPR=GPR with categorical variable, AR1=Autoregressive co-Kriging)

| **Name** | **Ref.** | **Type** | **Common mesh** | **DR** | **Intermediate surrogate** |
|---|---|---|---|---|---|
| S-HFPCA-GPR | – | – | – | PCA of $\mathcal{Y}_1$ | GP regression |
| TensCovGPR | [38] | – | – | – | GP regression |
| C-DiffPCA-GPR | [4] | Corrective | Yes | PCA of $\mathcal{Y}_\Delta$ | GP regression |
| M-GPCA | [9] | Mapping | No | Gappy-PCA | Gappy-PCA |
| M-SFPCA-GPR | [10, 73] | Mapping | No | Independent PCA of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ | GP regression |
| F-CPCA-AR1 | [1] | Fusion | Yes | Adapted constrained PCA | AR1 co-Kriging |
| F-HFPCA-AR1 | [8] | Fusion | Yes | PCA of $\mathcal{Y}_1$ | AR1 co-Kriging |
| F-MFPCA-AR1 | [7] | Fusion | Yes | PCA of $\mathcal{Y}_1 \cup \mathcal{Y}_2$ | AR1 co-Kriging |
| F-MFPCA-CategGPR | [11] | Fusion | Yes | PCA of $\mathcal{Y}_1 \cup \mathcal{Y}_2$ | GP regression with a categorical variable indicating the fidelity |
| F-SFPCA-AR1 | [2] | Fusion | No | Independent PCA of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ | AR1 co-Kriging |
| F-SFPCA-MA-AR1 | [5] | Fusion | No | Independent PCA of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ with MA | AR1 co-Kriging |
| F-SFKPCA-MA-AR1 | [6] | Fusion | No | Independent Kernel PCA of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ with MA | AR1 co-Kriging |
| F-SFLTSA-MA-AR1 | [6] | Fusion | No | Independent LTSA of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ with MA | AR1 co-Kriging |
| F-SFIsomap-MA-AR1 | [6] | Fusion | No | Independent Isomap of $\mathcal{Y}_1$ and $\mathcal{Y}_2$ with MA | AR1 co-Kriging |
| F-LFPCA-AR1-Resid | [3] | Fusion | Yes | PCA of $\mathcal{Y}_2$ | AR1 co-Kriging and Kriging with tensorized covariance |
| F-LFCVBPCA-AR1-Resid | [3] | Fusion | Yes | Cross-validation based PCA of $\mathcal{Y}_2$ | AR1 co-Kriging and Kriging with tensorized covariance |

$$e^{\text{norm}} = \frac{e}{\sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left|\left|\mathbf{y}_1^{(i)} - \bar{\mathbf{y}}_1\right|\right|^2}}. \tag{21}$$

For fusion-based surrogates using linear DR, the rms error can be decomposed to highlight the contributions of DR and of the intermediate surrogate, similarly to Perron *et al.* [74]. First, the DR error is computed by evaluating the rms of the difference between the validation snapshots and the validation snapshots mapped to the latent space and then mapped back. Hence, this error corresponds to the information lost by DR and its inverse mapping.

$$e_{\text{dr}} = \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left|\left|\mathbf{y}_1^{(i)} - \mathcal{DR}_1^{-1}\left(\mathcal{DR}_1\left(\mathbf{y}_1^{(i)}\right)\right)\right|\right|^2} \tag{22}$$

This DR rms error can be normalized as in Equation (21),

$$e_{\text{dr}}^{\text{norm}} = \frac{e_{\text{dr}}}{\sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left|\left|\mathbf{y}_1^{(i)} - \bar{\mathbf{y}}_1\right|\right|^2}} \tag{23}$$

Then, the intermediate surrogate modeling error is obtained by performing the following steps. First, the difference between the exact value of the latent variables obtained by applying DR to the validation snapshots $\mathcal{DR}_1\left(\mathbf{y}_1^{(i)}\right)$ and their prediction by the surrogate is computed for each sample $\hat{\mathbf{z}}_1^{(i)} = \mathcal{IS}_{\text{MF}}\left(\mathbf{u}_l^{(i)}\right)$. This difference cannot be computed for the case of nonlinear DR because they are not vectors but points on a manifold. Then, the difference is mapped back through the DR technique for each sample. Finally, the rms is computed,

$$e_{\text{ism}} = \sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left|\left|\mathcal{DR}_1^{-1}\left(\mathcal{DR}_1\left(\mathbf{y}_1^{(i)}\right) - \hat{\mathbf{z}}_1^{(i)}\right)\right|\right|^2}. \tag{24}$$

Once again, the intermediate surrogate modeling rms error can be normalized as in Equations (21) and (23):

$$e_{\text{ism}}^{\text{norm}} = \frac{e_{\text{ism}}}{\sqrt{\frac{1}{n_v}\sum_{i=1}^{n_v}\left|\left|\mathbf{y}_1^{(i)} - \bar{\mathbf{y}}_1\right|\right|^2}}. \tag{25}$$

For the specific case of PCA, the relation between the rms error, the DR and the intermediate surrogate modeling error sum up, as shown in [74]:

$$e^2 = e_{\text{dr}}^2 + e_{\text{ism}}^2. \tag{26}$$

These different errors allow to identify the most important contribution between DR and intermediate surrogate modeling to the global rms error. Additionally, the CPU-time required for training surrogates is recorded. Information about the hardware used can be found in Appendix A.

## 6. Results and discussions

The performance of the various multi-fidelity surrogates is now analyzed for the different test cases. The reading guide to the various plots is given with the results of the first test case in Section 6.1. The contribution of the different building blocks are then discussed. The numerical settings are detailed in Appendix A.

### *6.1. Viscous free fall of a ball without ground*

The results for the viscous free fall of a sphere without ground are shown in Figure 14. The benchmarked surrogates are listed along the vertical axis with their names given in Table 1. Let us recall that $n_1$ (resp. $n_2$) is the number of high-fidelity (resp. low-fidelity) samples. For each combination of ($n_1$, $n_2/n_1$, index of repetition), each surrogate is trained, its RMSE is measured as described in Section 5.2 and is ranked in ascending order of RMSE. Figure 14 displays a bar plot with the number of occurrences of each ranking being the length of the bars. For instance, the dark blue bar counts the number of times a given surrogate ranked best in terms of RMSE over all combinations of ($n_1$, $n_2/n_1$, index of repetition). Similarly, the dark red bar counts the number of times a given surrogate ranked worst. The colors between dark blue and dark red correspond to the ranking given in the legend of Figure 14 to the right. To avoid the bias of a "strict ranking", an additional metric is plotted in Figure 14, which is depicted by the scatterplot (dots, squares and triangles). It allows us to measure how far apart surrogates are from the best performing one. For each combination of ($n_1$, $n_2/n_1$, index of repetition) the lowest RMSE of the surrogates is taken as a reference. Then, for each surrogate, the number of times that their RMSE is lower than 1.05 times the reference RMSE is counted. This number is represented by the dots and is used to sort the surrogates vertically (the first one being the best and the last one the worst). The procedure is the same for the square and triangle points, which accounts the number of times a RMSE is lower than 1.25 and 2 times the reference RMSE, respectively. This visualization

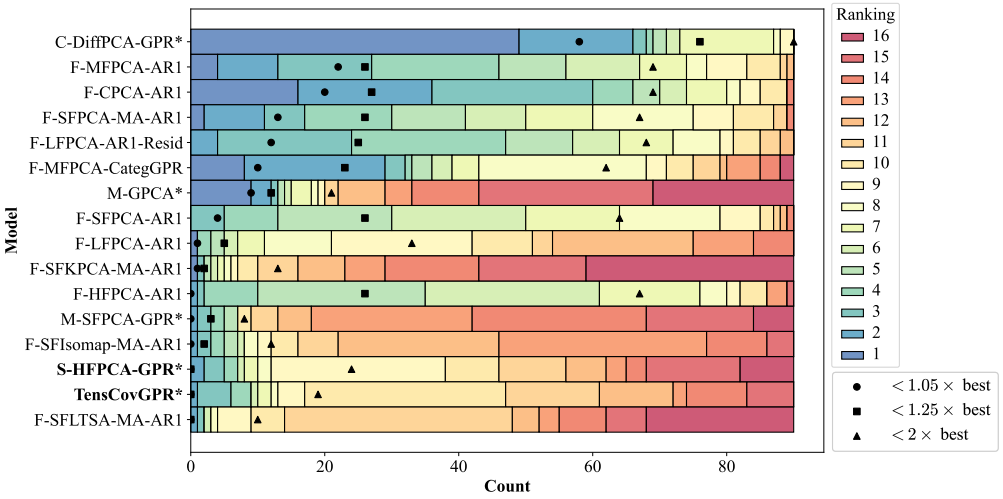



Figure 14: Global ranking of the various surrogates for the viscous free fall of a ball without ground. The colored bars show the ranking of the multi-fidelity surrogates. They are sorted vertically in decreasing number of times their normed RMSE is less than 5%, 25% and 100% away from the best performing multi-fidelity surrogate.

is inspired by the work of Lüthen *et al.* [75]. Similar graphs are shown in the remainder of the paper for different test cases.

Figure 14 shows that on this first case study, the corrective approach C-DiffPCA-GPR performs much better than any other method (it is ranking in the top 3 in 58 repetitions out of 90). A sort of plateau of fusion surrogates performing similarly can be seen, including F-MFPCA-AR1, F-CPCA-AR1, F-LFPCA-AR1-Resid, F-SFPCA-MA-AR1 and F-MFPCA-CategGPR, corresponding to rankings $2^{\text{nd}}$ to $6^{\text{th}}$. The best performing fusion surrogate is F-MFPCA-AR1. Mapping methods, on the contrary, are among the least accurate ones. Even if well ranked, M-GPCA has a large proportion for lower rankings (dark red bar, 67 repetitions over 90 among the 3 worst ranks), meaning that this method is unreliable in this particular test case. Every multi-fidelity surrogate using linear DR is performing better than both reference single-fidelity surrogates.

When comparing F-SFPCA-MA-AR1 and F-SFPCA-AR1, respectively ranked $4^{\text{th}}$ and $8^{\text{th}}$, it appears that manifold alignment brings a significant improvement. The comparison between fusion surrogates that only vary by their DR technique (F-MFPCA-AR1, F-SFPCA-AR1, F-HFPCA-AR1, F-LFPCA-AR1 and F-CPCA-AR1) shows that using MFPCA or CPCA yields

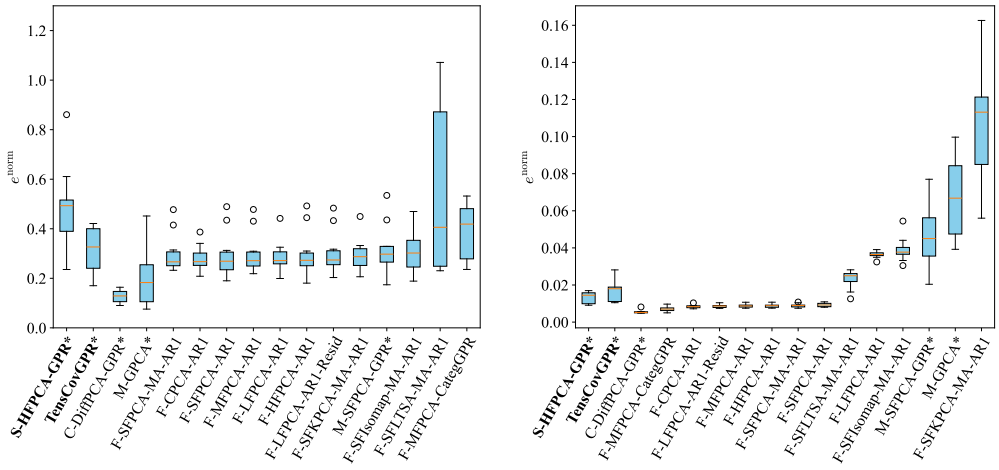



Figure 15: Boxplot of the normed RMSE for the viscous free fall without ground case study with $n_1 = 8$ and $n_2 = 8$ (left) and $n_1 = 40$ and $n_2 = 400$ (right). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

significantly better results for this test case, with respectively 22 and 20 runs with a RMSE less than 5% higher than the lowest RMSE, compared to 4 for the best of the others. As for the intermediate surrogate, the comparison of F-MFPCA-AR1 and F-MFPCA-CategGPR shows AR1 is a better choice for this test case, as F-MFPCA-CategGPR only performs 10 times less than 5% worse than the best performing surrogate. Furthermore, F-LFPCA-AR1-Resid is ranked $5^{\text{th}}$ while F-LFPCA-AR1 is ranked $9^{\text{th}}$, illustrating a probable interest of the taking into account the dimensionality reduction residuals.

Ranking surrogates as in Figure 14 yields a hierarchy between them. However, it does not show how fart apart surrogates are. The RMSE of the different surrogates are displayed in Figure 15. The two extreme cases $n_1 = 2 \times \dim(U)$ with $n_2 = n_1$ and $n_1 = 10 \times \dim(U)$ with $n_2 = 10 \times n_1$ are shown. The two single-fidelity surrogates (*i.e.*, S-HFPCA-GPR and TensCovGPR) are stacked on the left for reference and multi-fidelity surrogates are sorted by increasing media RMSE. This graphical representation provides additional context to the rankings. When data is scarce, $n_1 = 8$ and $n_2 = 8$, the lowest median normed RMSE is reached by C-DiffPCA-GPR with 12.9%. The largest median normed RMSE is reached by the single-fidelity surrogate S-HFPCA-GPR with 49.3%. The plateau of multi-fidelity surrogates is apparent but is different from the one seen in the rankings since results are not

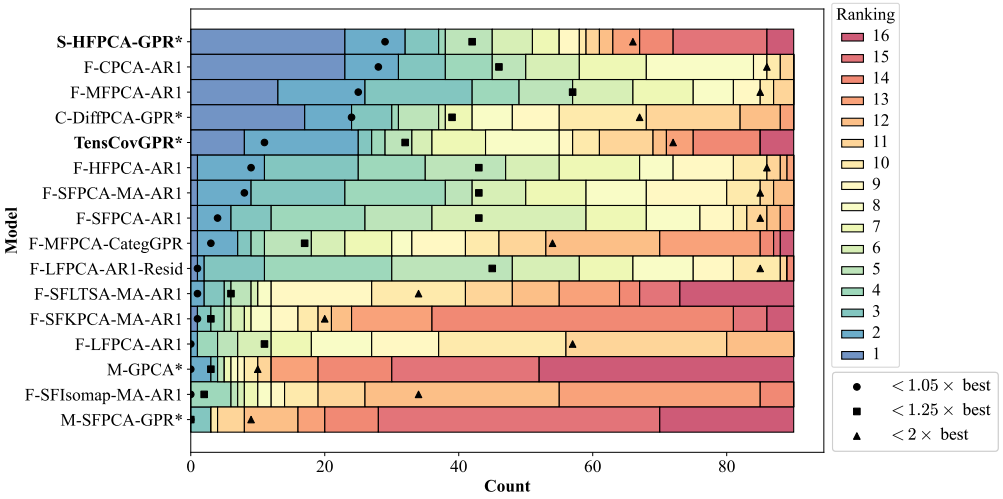



Figure 16: Global ranking of the various surrogates for the viscous free fall of a ball with ground. The colored bars show the ranking of the multi-fidelity surrogates. They are sorted vertically in decreasing number of times their normed RMSE is less than 5%, 25% and 100% away from the best performing multi-fidelity surrogate.

aggregated anymore.

When data is abundant, $n_1 = 40$ and $n_2 = 400$, the lowest median normed RMSE is reached again by C-DiffPCA-GPR with 0.5% while the largest is reached by F-SFKPCA-MA-AR1 with 11.3%. Some multi-fidelity surrogates that were close to the others are now falling behind, including surrogates using nonlinear DR, mapping surrogates and F-LFPCA-AR1. Note that mapping surrogates do not benefit from the addition of low-fidelity snapshots that do not have corresponding high-fidelity snapshots. Surrogates using nonlinear DR seem to benefit less from the addition of snapshots than surrogates using linear DR. Anyhow, every surrogate is performing better with the addition of high- and low-fidelity snapshots except F-SFKPCA-MA-AR1.

### *6.2. Viscous free fall of a ball with ground*

Figure 16 shows the results for the viscous free fall with a ground test case. Simply adding a ground (and therefore a discontinuity in the snapshots) significantly changes the results. The best surrogate is now the single-fidelity surrogate S-HFPCA-GPR with 37 repetitions over 90 among the top 3 surrogates, but it has a high proportion of lower rankings (23 repetitions over 90 among the 3 worst ranks), meaning that it is particularly unreliable on

this test case. It is closely followed by the fusion surrogate F-CPCA-AR1 coming second, which has no repetition among the 3 worst surrogates. One might prefer to use a lower ranking multi-fidelity surrogate instead of the mono-fidelity surrogate. The corrective C-DiffPCA-GPR surrogate (ranked $4^{\text{th}}$) which was ranked best before the addition of the ground is now ranking lower than the single-fidelity surrogate S-HFPCA-GPR (ranked $1^{\text{st}}$) and the best fusion surrogate (ranked $2^{\text{nd}}$). The corrective surrogate has a moderately high proportion of bad performance as is shown by the triangle marks. Mapping surrogates (M-GPCA and M-SFPCA-GPR) are performing much worse than other multi-fidelity and the single-fidelity surrogates on this test case. The addition of manifold alignment is now leading to a smaller improvement as can be seen by comparing F-SFPCA-MA-AR1 (ranked $7^{\text{th}}$) and F-SFPCA-AR1 (ranked $8^{\text{th}}$). Comparing the fusion surrogates that only vary by their DR technique (F-CPCA-AR1, F-MFPCA-AR1, F-SFPCA-AR1, F-HFPCA-AR1 and F-LFPCA-AR1) shows that CPCA and MFPCA performs significantly better than the best of the others, with respectively 28 and 25 against 9 repetitions with a good RMSE (where good means a RMSE below the best RMSE plus 5%). Finally, the comparison of F-MFPCA-AR1 (ranked $3^{\text{rd}}$) and F-MFPCA-CategGPR (ranked $9^{\text{th}}$) shows that AR1 is a better multi-fidelity approach for this test case.

The normed RMSE are plotted in Figure 17. In the scarce data context, $n_1 = 8$ and $n_2 = 8$, the lowest median normed RMSE is reached by C-DiffPCA-GPR with 13.9% while the largest is 49.3% for S-HFPCA-GPR. Once again, a plateau can be seen, covering most multi-fidelity surrogates, even though the ones with a larger median normed RMSE have a much larger variability. With more data to learn, when $n_1 = 40$ and $n_2 = 400$, the lowest median normed RMSE is 1.3% for F-MFPCA-AR1 while the largest is 16.5% for M-GPCA. This time, single-fidelity surrogates are performing similarly to the best multi-fidelity surrogates with a normed RMSE of 1.4% for S-HFPCA-GPR and 1.7% for TensCovGPR. Again, mapping surrogates benefit less than other surrogates from the addition of snapshots. Surrogates using nonlinear DR, F-LFPCA-AR1, F-MFPCA-CategGPR and C-DiffPCA-GPR also benefit less.

### *6.3. Pressure coefficient field around a NACA 0015 airfoil*

Figure 18 depicts the results for the test case of a one-dimensional pressure coefficient field around NACA 0015 airfoils. C-DiffPCA-GPR is now the best performing method, with 47 repetitions out of 90 ranking in the top 3

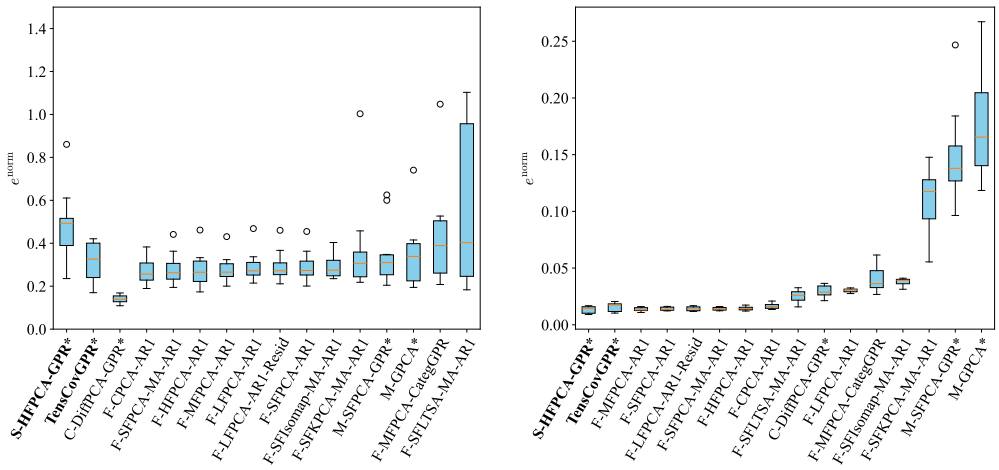



Figure 17: Boxplot of the normed RMSE for the viscous free fall with a ground case with $n_1 = 8$ and $n_2 = 8$ (left) and $n_1 = 40$ and $n_2 = 400$ (right). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

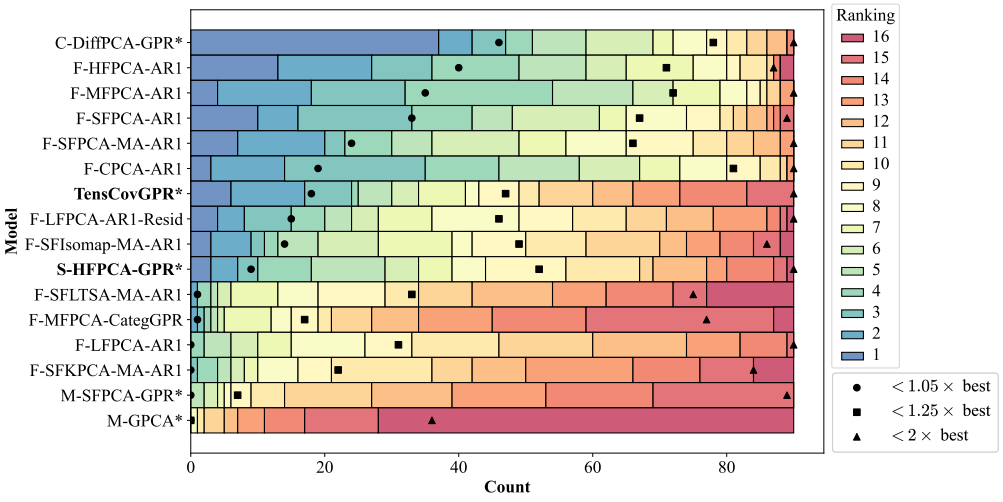



Figure 18: Global ranking of the various surrogates for the NACA 0015 airfoil. The colored bars show the ranking of the multi-fidelity surrogates. They are sorted vertically in decreasing number of times their normed RMSE is less than 5%, 25% and 100% away from the best performing multi-fidelity surrogate.

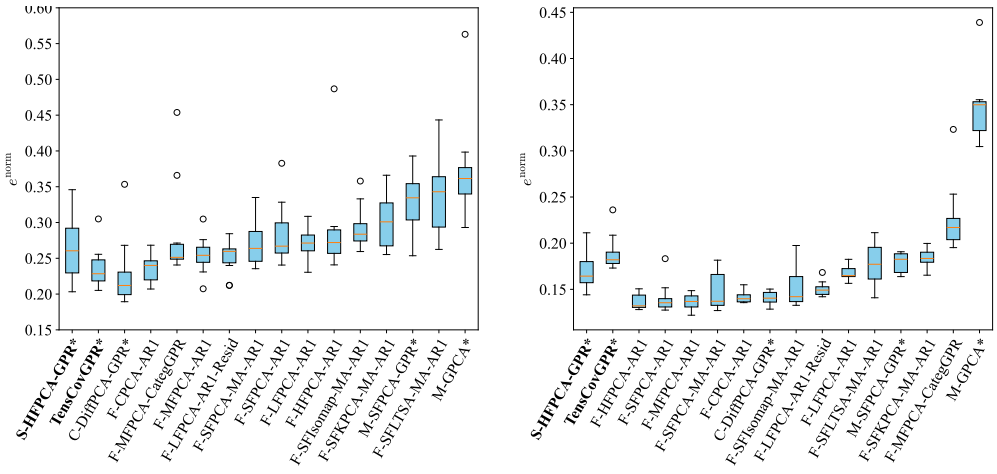



Figure 19: Boxplot of the normed RMSE for the NACA 0015 airfoil case with $n_1 = 10$ and $n_2 = 10$ (left) and $n_1 = 50$ and $n_2 = 500$ (right). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

surrogates. It is followed by the fusion surrogate F-HFPCA-AR1, with 36 repetitions in the top 3. The corrective surrogate and most of the fusion surrogates perform better than S-HFPCA-GPR. Surrogates using LFPCA, nonlinear DR, CategGPR or the mapping approach all rank lower than TensCovGPR (ranked $7^{\text{th}}$). The mapping surrogates are the worst performing surrogates on this test case, with 51 and 79 repetitions out of 90 in the 3 worst ranking surrogates. Here, comparing F-SFPCA-MA-AR1 and F-SFPCA-AR1 shows that the use of manifold alignment leads to a small decrease in the performance. F-MFPCA-AR1, F-SFPCA-AR1 and F-HFPCA-AR1 have very similar performances, suggesting that the choice of DR between MFPCA, SFPCA and HFPCA has no particular impact when combined with AR1 co-Kriging for this test case. As for the comparison of F-MFPCA-AR1 (ranked $3^{\text{rd}}$) and F-MFPCA-CategGPR (ranked $12^{\text{th}}$), this shows again the superiority of AR1 over CategGPR.

The normed RMSE are plotted in Figure 19. When data is scarce, $n_1 = 10$ and $n_2 = 10$, the lowest median normed RMSE is 21.2% for C-DiffPCA-GPR while the largest is reached by M-GPCA with 36.1%. Note that when both high- and low-fidelity snapshots are scarce, single fidelity-surrogates perform similarly to multi-fidelity surrogates. When there is much learning data, $n_1 = 50$ and $n_2 = 500$, the lowest median normed RMSE is reached by F-HFPCA-

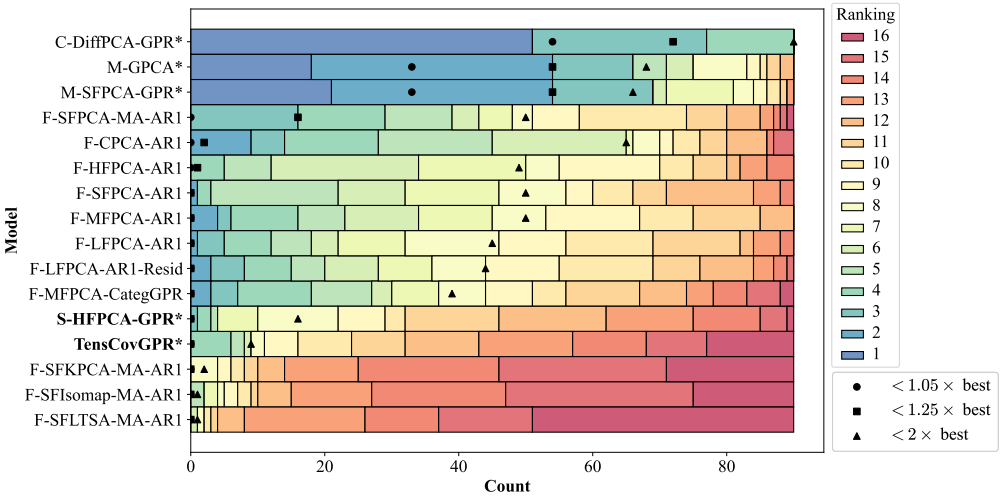



Figure 20: Global ranking of the various surrogates for the RAE 2822 airfoil, with the CFD RANS simulator with a fine mesh as the high-fidelity simulator, and the CFD RANS with a coarse mesh as the low-fidelity simulator. The colored bars show the ranking of the multi-fidelity surrogates. They are sorted vertically in decreasing number of times their normed RMSE is less than 5%, 25% and 100% away from the best performing multi-fidelity surrogate.

AR1 with 13.2% while the largest is reached by M-GPCA with 35.0%. Note that M-GPCA barely benefits from the addition of snapshots. With more high- and low-fidelity snapshots, multi-fidelity surrogates outperform their single-fidelity counterparts.

### *6.4. Pressure coefficient field around the RAE 2822 airfoil (RANS/RANS configuration)*

Figure 20 shows the results for the RAE 2822 airfoil test case with the CFD RANS simulator with a fine mesh as the high-fidelity simulator and the CFD RANS with a coarse mesh as the low-fidelity simulator. Here, the corrective surrogate performs best. It is always in the top 3 and is ranking $1^{\text{st}}$ in 51 repetitions out of 90, showing a high robustness. The two mapping surrogates are ranking $2^{\text{nd}}$ and $3^{\text{rd}}$ and are performing very similarly, with 66 and 69 repetitions in the top 3. These three surrogates yield better results than the single-fidelity surrogates and all of the fusion surrogates, as none of them manages a RMSE within 5% of the best RMSE. The histogram shows that only the corrective and mapping surrogates manage to

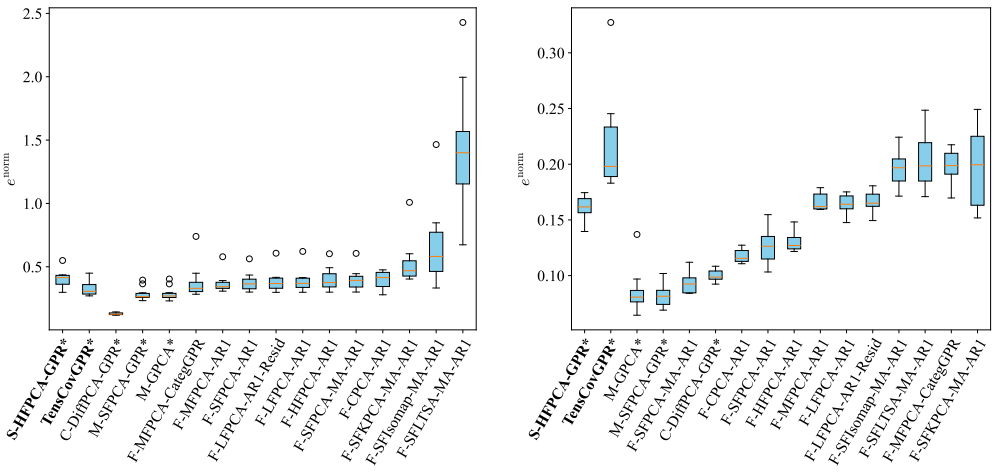



Figure 21: Boxplot of the normed RMSE for the RAE 2822 RANS/RANS case with $n_1 = 6$ and $n_2 = 6$ (left) and $n_1 = 30$ and $n_2 = 300$ (right). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

rank first for a combination of ($n_1$, $n_2/n_1$, index of repetition). The best performing fusion surrogate is F-SFPCA-MA-AR1 with 16 repetitions and a RMSE within 25% of the best RMSE. Most fusion surrogates with linear DR have relatively similar performances. The worst surrogates for this test case are the single-fidelity surrogates and multi-fidelity surrogates using nonlinear DR. As for manifold alignment, it slightly improves the results (rank $4^{\text{th}}$ of F-SFPCA-MA-AR1 versus rank $7^{\text{th}}$ of F-SFPCA-AR1). F-CPCA-AR1, F-MFPCA-AR1, F-SFPCA-AR1 and F-HFPCA-AR1 surrogate models are performing very similarly, as is F-LFPCA-AR1 (ranked $5^{\text{th}}$ to $9^{\text{th}}$). Finally, the comparison of F-MFPCA-AR1 (ranked $8^{\text{th}}$) and F-MFPCA-CategGPR (ranked $11^{\text{th}}$) shows that AR1 is a better intermediate surrogate, even though this is less pronounced than for the previous test cases.

The boxplots of the normed RMSE for this test case are displayed in Figure 21. When $n_1 = 6$ and $n_2 = 6$, the lowest median normed RMSE is 12.6% for C-DiffPCA-GPR while the largest is reached by F-SFLTSA-MA-AR1 with 140.0%. There are again two groups of surrogates: the multi-fidelity surrogates which perform better, to the exception of C-DiffPCA-GPR, and surrogates using nonlinear DR which perform worse. Single-fidelity surrogates are performing similarly to multi-fidelity surrogates that belong to the plateau. With more data ($n_1 = 30$ and $n_2 = 300$), the lowest median normed

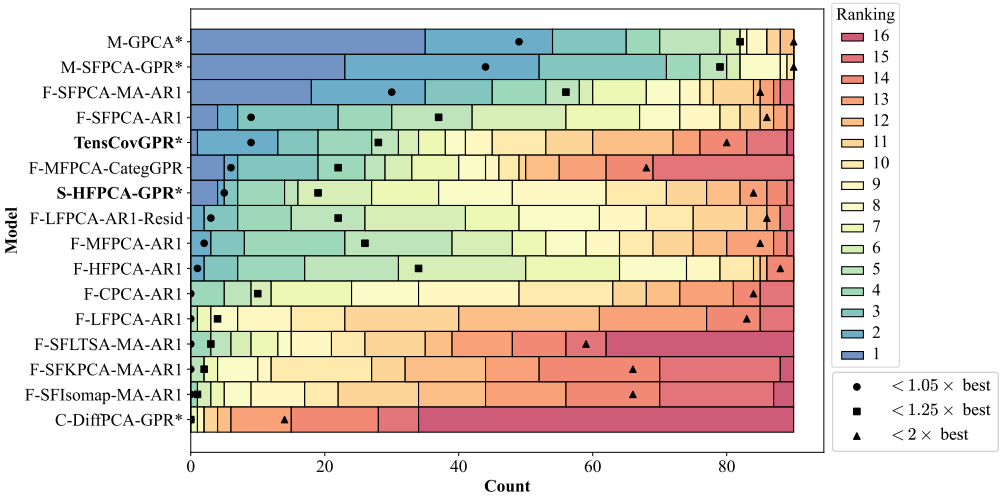



Figure 22: Global ranking of the various surrogates for the RAE 2822 airfoil, with the CFD RANS simulator with a fine mesh as the high-fidelity simulator and the CFD Euler with a coarse mesh as the low-fidelity simulator. The colored bars show the ranking of the multi-fidelity surrogates. They are sorted vertically in decreasing number of times their normed RMSE is less than 5%, 25% and 100% away from the best performing multi-fidelity surrogate.

RMSE is 8.1% for M-GPCA while the largest is 20.0% for F-SFKPCA-AR1. There is no longer a plateau and single-fidelity surrogates are now performing significantly worse than the best multi-fidelity surrogates.

### *6.5. Pressure coefficient field around the RAE 2822 airfoil (RANS/Euler configuration)*

The results for the RAE 2822 airfoil are depicted in Figure 22 where the high-fidelity simulator is the CFD RANS simulator with a fine mesh and the CFD Euler with a coarse mesh is the low-fidelity simulator. The best performing methods are the mapping surrogates with 65 repetitions or more among the top 3 surrogates. They have similar results even though M-GPCA is slightly better. Next, the best performing fusion surrogate is F-SFPCA-MA-AR1, with 45 repetitions in the top 3. Not many fusion surrogates perform better than the reference single-fidelity surrogates. F-MFPCA-CategGPR is ranking among the average fusion surrogates but has a high proportion of low rankings (28 repetitions out of 90 among the 3 worst surrogates). Finally, the corrective approach is performing worse than any

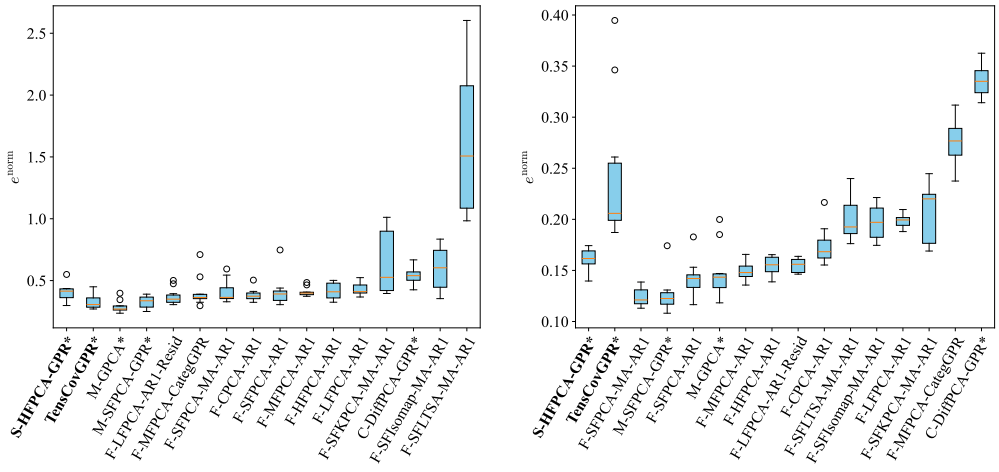



Figure 23: Boxplot of the normed RMSE for the RAE 2822 RANS/Euler case with $n_1 = 6$ and $n_2 = 6$ (left) and $n_1 = 30$ and $n_2 = 300$ (right). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

other surrogate with 75 repetitions out of 90 among the 3 worst surrogates. Its performance is further discussed in Section 6.7.4. Comparing F-SFPCA-MA-AR1 (ranked 3$^{\text{rd}}$) and F-SFPCA-AR1 (ranked 4$^{\text{th}}$) shows a noticeable improvement due to manifold alignment. Unlike in any of the previous test cases, F-MFPCA-CategGPR (ranked 6$^{\text{th}}$) is ranked higher than F-MFPCA-AR1 (ranked 9$^{\text{th}}$). But as said previously, F-MFPCA-CategGPR has a high proportion of low rankings.

To complement the ranking information, the boxplots of the normed RMSE are displayed in Figure 23. With little data to learn, $n_1 = 6$ and $n_2 = 6$, the lowest median normed RMSE is 27.0% for M-GPCA while the largest is reached by F-SFLTSA-MA-AR1 with 150.7%. Most multi-fidelity surrogates have a similar median normed RMSE except for surrogates using nonlinear DR and C-DiffPCA-GPR. Single-fidelity surrogates have a performance similar to multi-fidelity surrogates. When $n_1 = 30$ and $n_2 = 300$, the lowest median normed RMSE is 12.1% for F-SFPCA-MA-AR1 while the largest is 33.5% for C-DiffPCA-GPR. Single-fidelity surrogates tend to lag behind multi-fidelity surrogates in terms of median normed RMSE. Additionally, there is no noticeable plateau anymore.

In this test case, the additional low-fidelity data leads to little or no improvement in the prediction capabilities of multi-fidelity surrogates. Indeed,

the best performing models are mapping surrogates. This is likely due to the poor correlation between the high- and low-fidelity fields.

### *6.6. Computational training cost*

The computational training cost in terms of CPU-time of the different surrogates across the different test cases are displayed in Figure 24. This CPU-time does not include the simulation time needed to get the training samples. Added to the fact that the DoEs have the same size, the given computational costs are direct translations of the computational complexities of the surrogate models. To ensure a fair comparison, the different surrogates are using similar implementations (see Appendix A for more details on the chosen libraries and the tuning of hyperparameters). The training CPU-time is averaged across all combinations of ($n_1$, $n_2/n_1$, index of repetition).

It can be seen that the NACA 0015 airfoil case incurs a significantly larger training cost than all other test cases. This discrepancy suggests that the training computational cost is not only influenced by the dimensionality of the fields. Indeed, the order of magnitude of the number of nodes in the NACA 0015 is $10^3$, while it is $10^4$ in the RAE 2822 airfoil. In this benchmark, one of the main contributors to the overall training computational cost is the dimensionality of the latent space. In fact, one surrogate per latent variable must be built and trained. The dimensionality latent space of the NACA 0015 is larger than that of the RAE 2822 airfoil. Regarding the comparison of the training computational cost of the different surrogates, the M-GPCA surrogate has the lowest training computational cost (lower than a second on average for all test cases). This efficiency can be attributed to the fact that building GPCA involves only a few linear algebra operations with one hyperparameters that needs tuning, unlike the surrogates using GP-based regression (*i.e.*, GPR, AR1 and CategGPR). The single-fidelity surrogate TensCovGPR is the second fastest surrogate. Then, there are multi-fidelity surrogates using Isomap and LTSA. This can be explained by the fact that the dimensionality of the latent space is small, leading to the training of few intermediate surrogates (see Appendix A for more details about the numerical settings). C-DiffPCA-GPR has a training computational cost similar to the reference single-fidelity surrogate. Subsequently, the next multi-fidelity surrogate incurs a computational cost approximately 4.5 times larger than the reference single-fidelity surrogate. The longest multi-fidelity surrogate has an average training CPU-time about 64 times larger than the single-fidelity surrogate. F-SFPCA-MA-AR1 is slightly faster to train than

F-SFPCA-AR1. Manifold alignment involves only a few linear algebra operations in the low-dimensional latent space, resulting in a negligible additional computational cost. This suggests that manifold alignment, by increasing the correlation between the high- and low-fidelity data, accelerates the training of the multi-fidelity intermediate surrogate model. The small increase in training time of the F-LFPCA-AR1-Resid over the F-LFPCA-AR1 except for the NACA 0015 airfoil indicates that the modeling of the DR residuals Kriging with tensorized covariance should usually not be too costly.

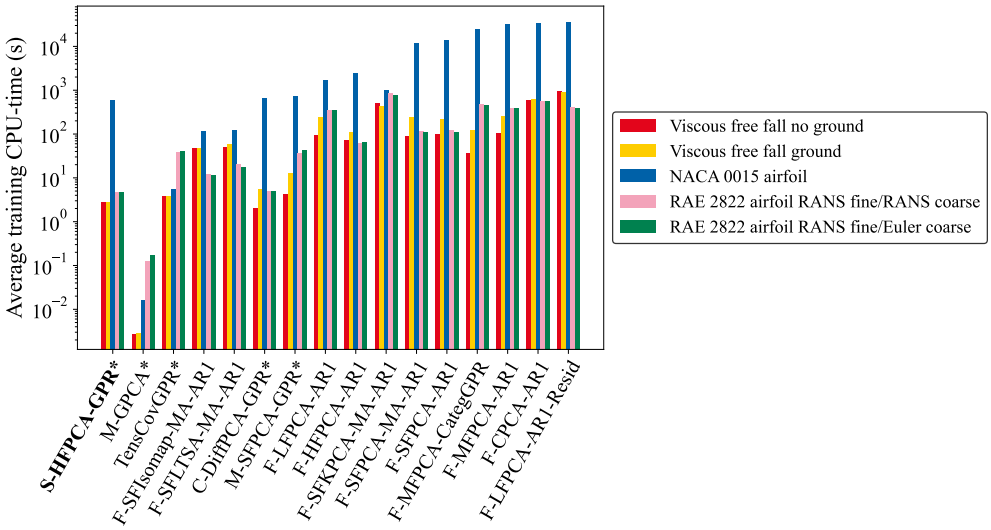



Figure 24: Histogram of the training CPU-time in seconds. The colors of the bars correspond to the different tests cases. Details about the implementation and hardware used can be found in Appendix A.

### *6.7. Discussions*

#### *6.7.1. Single- versus multi-fidelity surrogates*

Before delving into the specificities of the different multi-fidelity surrogates, we assess if adopting a multi-fidelity scheme is beneficial. In the Figures 14, 16, 18, 20 and 22, single-fidelity surrogates only rank first once. S-HFPCA-GPR ranks $14^{\text{th}}$, $1^{\text{st}}$, $10^{\text{th}}$, $12^{\text{th}}$ and $7^{\text{th}}$, while TensCovGPR ranks $15^{\text{th}}$, $5^{\text{th}}$, $7^{\text{th}}$, $13^{\text{th}}$ and $5^{\text{th}}$. Note that no single-fidelity surrogate seem to significantly outperform the other. In Figures 15, 17, 19, 21 and 23, when data is scarce (in the sense that both high- and low-fidelity data is limited or the

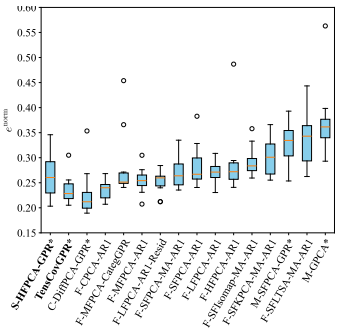
high- and low-fidelity fields are poorly correlated), single-fidelity surrogates perform similarly to multi-fidelity surrogates.

Boxplots of the normed RMSE for different test cases are displayed in Figure 25 with a constant number of high-fidelity snapshots $n_1$ and a varying number of low-fidelity snapshots $n_2$. The first row is the NACA 0015 airfoil case with $n_1 = 2 \times \dim(U)$, the second row is the RAE 2822 airfoil RANS/RANS case with $n_1 = 5 \times \dim(U)$ and the third row is the RAE 2822 airfoil RANS/Euler case with $n_1 = 10 \times \dim(U)$. The left column is for $n_2 = n_1$ while the second column is for $n_2 = 10 \times n_1$.

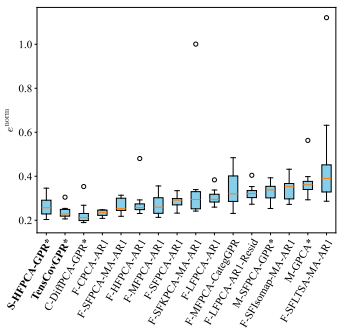
Single-fidelity, mapping and corrective surrogates do not benefit from the addition of low-fidelity snapshots for a fixed number of high-fidelity snapshots. In the NACA 0015 airfoil case with a small number of high-fidelity snapshots ($n_1 = 2 \times \dim(U)$), the addition of low-fidelity snapshots only marginally improves the performance of multi-fidelity surrogates. For the RAE 2822 RANS/RANS airfoil case for the intermediate number of high-fidelity snapshots ($n_1 = 5 \times \dim(U)$), adding low-fidelity snapshots significantly improves the performance of multi-fidelity surrogates combining linear DR (except LFPCA) with AR1 co-Kriging. It appears that surrogates using nonlinear DR, LFPCA or CategGPR are hardly affected by the data augmentation. For the RAE 2822 RANS/Euler airfoil case, the improvement on the performance is only marginal. It even worsen the results for the multi-fidelity surrogate using CategGPR as the intermediate surrogate. Nevertheless, multi-fidelity surrogates perform better than single-fidelity surrogates, even though the addition of low-fidelity snapshots for a fixed number of high-fidelity snapshot only marginally improves the results. Note that the benefit of the addition of low-fidelity data highly depends on the correlation between the high- and low-fidelity fields. In particular, the RAE 2822 Euler data is, by construction, much less informative of the high-fidelity than the RANS with coarse mesh data.

*6.7.2. Linear versus nonlinear dimension reduction*

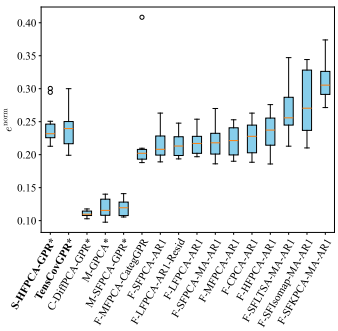
A method to assess the effectiveness of linear DR is to measure the relative information content (RIC) of the PCA. Figure 26 illustrates the rate of decay of 1 - RIC of a PCA on 1,000 high-fidelity snapshots for each test case. Note that the PCA for the viscous free fall cases can have a maximum of 101 non-zero eigenvalues which is the number of nodes in the mesh, even though the number of samples exceeds the number of nodes. Consequently, their PCAs exhibit a faster decay in 1 - RIC compared to the other test cases.



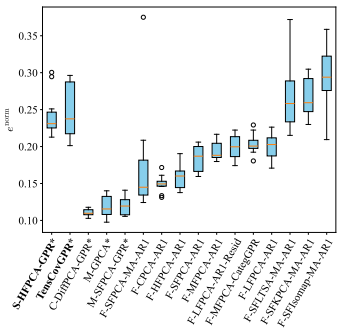

(a) NACA 0015, $n_1 = 10$ and $n_2 = 10$



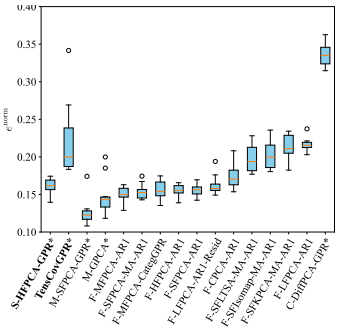

(b) NACA 0015, $n_1 = 10$ and $n_2 = 100$



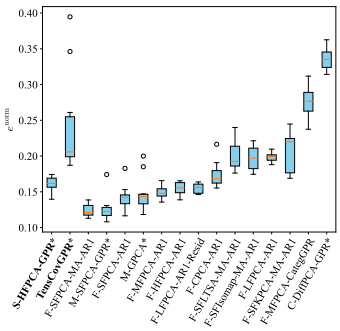

(c) RAE 2822 RANS/RANS, $n_1 = 15$ and $n_2 = 15$




(d) RAE 2822 RANS/RANS, $n_1 = 15$ and $n_2 = 150$




(e) RAE 2822 RANS/Euler, $n_1 = 30$ and $n_2 = 30$




(f) RAE 2822 RANS/Euler, $n_1 = 30$ and $n_2 = 300$

Figure 25: Boxplots of the normed RMSE for the test cases with $n_2 = n_1$ (left column) and $n_2 = 10 \times n_1$ (right column). Single-fidelity surrogates (bold label) are stacked to the left and multi-fidelity surrogates are sorted by median normed RMSE.

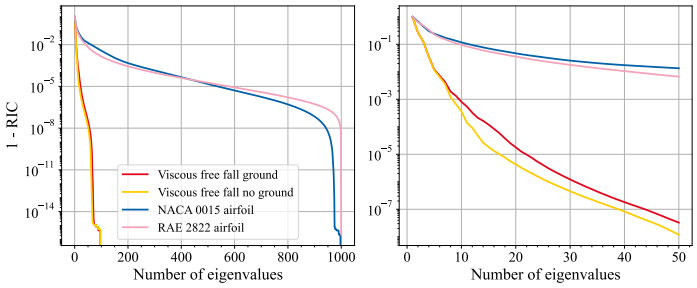



Figure 26: Rate of decay of 1 - RIC of a PCA on the 1,000 high-fidelity validation snapshots for each test case. Zoom on the 50 first eigenvalues on the right. Since the high-fidelity simulator is the same for the two RAE 2822 configurations, only one rate of decay is shown for this case (pink solid line).

In contrast, the decay of the NACA 0015 and the RAE 2822 test cases is slower, indicating that a significant number of modes is required to perform a linear DR without a substantial information loss. For instance, for a relative information content of 99.9%, 8 modes are required for the viscous free fall without ground, 9 modes for the viscous free fall with ground, 155 modes for the NACA 0015 airfoil, and 115 modes for the RAE 2822 airfoil. These numbers highlight that the seemingly less complex NACA 0015 airfoil poses a greater challenge for linear DR than the RAE 2822 airfoil does.

In our experiments, the NACA 0015 and RAE 2822 airfoil cases did not benefit from nonlinear DR. The few multi-fidelity surrogates utilizing nonlinear DR, which are just variants of the surrogate of Perron *et al.* [5], do not outperform surrogates employing linear DR. Given the good performance of F-SFPCA-MA-AR1, this cannot be attributed to the assembly method of the surrogate F-SF{PCA, KPCA, Isomap, LTSA}-MA-AR1. Hence, there are two possible explanations for the inferior performance of nonlinear DR. First, the chosen techniques for nonlinear DR may not be suitable for the tested use cases. Second, KPCA, Isomap and LTSA require the selection of multiple hyperparameters, significantly complexifying the surrogate training process compared to PCA. Despite intensive tuning of the hyperparameters by the authors of this study, the performance of multi-fidelity surrogates using nonlinear DR did not surpass that of surrogates using linear DR. Exploring alternative nonlinear DR techniques could be a focus for future research.

Table 2: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for F-HFPCA-AR1, F-LFPCA-AR1 and F-MFPCA-AR1.

| Test case | surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|---|
| Viscous free fall | F-HFPCA-AR1 | 8.3 | 0.2 | 8.3 |
| without ground | F-LFPCA-AR1 | 9.7 | 3.7 | 8.1 |
| | F-MFPCA-AR1 | 8.2 | 0.1 | 8.2 |
| Viscous free fall | F-HFPCA-AR1 | 9.3 | 0.2 | 9.3 |
| with ground | F-LFPCA-AR1 | 10.3 | 3.1 | 9.4 |
| | F-MFPCA-AR1 | 9.4 | 0.1 | 9.4 |
| NACA 0015 | F-HFPCA-AR1 | 19.6 | 5.4 | 19.2 |
| | F-LFPCA-AR1 | 22.0 | 12.9 | 19.7 |
| | F-MFPCA-AR1 | 9.4 | 0.1 | 9.4 |
| RAE 2822 | F-HFPCA-AR1 | 22.2 | 14.7 | 17.1 |
| RANS/RANS | F-LFPCA-AR1 | 22.3 | 11.7 | 19.5 |
| | F-MFPCA-AR1 | 21.7 | 7.9 | 20.0 |
| RAE 2822 | F-HFPCA-AR1 | 25.2 | 14.7 | 22.1 |
| RANS/Euler | F-LFPCA-AR1 | 30.4 | 20.0 | 23.5 |
| | F-MFPCA-AR1 | 27.7 | 7.9 | 26.4 |

*6.7.3. Influence of the PCA type for PCA-based fusion surrogates*

Table 2 presents the normed RMSE, DR RMSE and intermediate surrogate modeling RMSE for the different test cases and for three different types of PCA: LFPCA, HFPCA and MFPCA. Recall that in LFPCA and HFPCA, the PCA is carried out on the low- or high-fidelity data, respectively, while MFPCA, the mixed-fidelity PCA, processes both data together. It appears that the DR error is lower with MFPCA than with LFPCA and HFPCA. While this does not guarantee superior prediction capabilities for the surrogate, it is a meaningful information. The rankings seen in Figures 14 to 23 show that F-MFPCA-AR1 generally yields better results than F-HFPCA-AR1 and F-LFPCA-AR1.

Let $\eta = n_2/(n_1 + n_2)$ be the proportion of low-fidelity snapshots used in the PCA: $\eta = 0$ in HFPCA, $\eta = 1$ in LFPCA, and it varies between 0 and 1 in MFPCA. Figure 27 gives the decay of 1-RIC for MFPCA, HFPCA and LFPCA in the viscous free fall with a ground and the RAE 2822 airfoil RANS/Euler test cases (see Appendix C for all test cases). Ten repetitions with different training snapshots are plotted, $n_1 = 40$ and $n_2 = 400$ in the viscous free fall with ground case, $n_1 = 30$ and $n_2 = 300$ in the RAE 2822 airfoil RANS/Euler case.

On the left (viscous free fall with ground), 1 - RIC for HFPCA ($\eta = 0$) and LFPCA ($\eta = 1$) decays faster than for intermediate value of $\eta$. On the right (RAE 2822 RANS/Euler), the decay of 1 - RIC slows as $\eta$ increases, meaning that there is more information in the first modes of HFPCA than in the first modes of LFPCA. Mixing the fidelities makes even more modes necessary in the PCA.

A risk with MFPCA that is inherent to mixing data from multiple fidelities, is that the distinctive features of the high-fidelity field may be blended into those of the low-fidelity field. Note that in general, in the context of multi-fidelity, it holds that $n_1 < n_2$. Depending on the respective computational cost of the high- and low-fidelity simulators, it can even be $n_1 \ll n_2$. When $n_2/n_1 \to \infty$, MFPCA converges to LFPCA.

*6.7.4. On the performance of the corrective approach*

This section delves further into the performance of the corrective approach (C-DiffPCA-GPR), which tends to rank among the top multi-fidelity surrogates. As this surrogate is based on learning the difference between high- and low-fidelity fields, some illustrative samples are presented for the different test cases in Appendix D. Recall that this surrogate model does

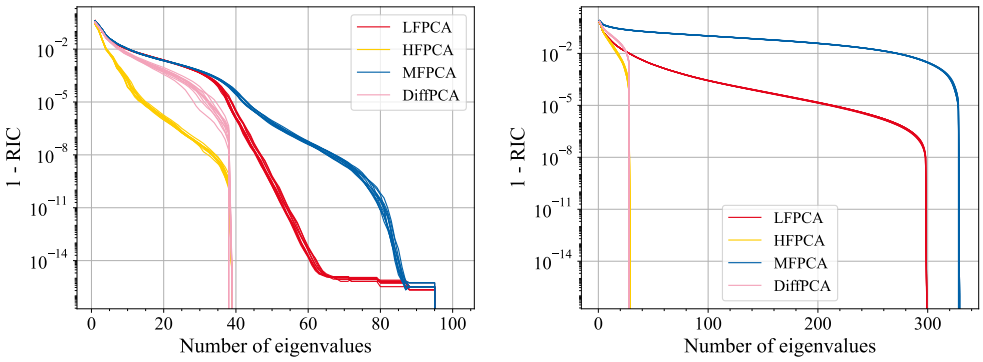



Figure 27: Decay of 1 - RIC for different PCAs for the viscous free fall with ground test case with $n_1 = 40$ and $n_2 = 400$ (left), and RAE 2822 airfoil RANS/Euler with $n_1 = 30$ and $n_2 = 300$ (right), 10 repetitions each

not benefit from the information provided by low-fidelity snapshots that do not have a corresponding high-fidelity snapshot. First, the viscous free fall problem is analyzed. The performance of the corrective surrogate is inferior when there is a ground (see Figure 16) than when there is not (see Figure 14). In the first scenario, it performs better than the reference single-fidelity surrogates but worse than S-HFPCA-GPR in the second. This can be understood by comparing the decay of 1 - RIC for DiffPCA and HFPCA computed from 200 high- and low-fidelity samples randomly selected in the validation set. For the viscous free fall with ground, the decay is comparable (see Figure 28a), while the addition of the ground results in a significantly slower decay of 1 - RIC for DiffPCA compared to HFPCA (see Figure 28b). This indicates that the low-dimensional representation of the difference between the high- and low-fidelity requires more principal components in the presence of the ground, likely due to the temporal displacement of the discontinuity caused by the impact on the ground which is much more present in low-fidelity snapshots. HFPCA, however, is barely affected by the addition of the ground since very little snapshots have a discontinuity, leading to the change in the ordering of the reference single-fidelity and the corrective surrogates. Note that even though we eventually use the same RIC threshold in both cases, this does not mean that the dimension reduction errors will be the same, thus the difference in prediction performance.

Shifting focus to the NACA 0015 airfoil, Figure 28c shows a similar decay of 1 - RIC for HFPCA and DiffPCA. Since the corrective surrogate leverages

the information provided by the low-fidelity simulator, it eventually outperforms the single-fidelity surrogate (see Figure 18). DiffPCA and HFPCA show a similar decay of 1 - RIC since the displacement of the boundary layer separation is present on both the high-fidelity and the difference between the high- and low-fidelity snapshots.

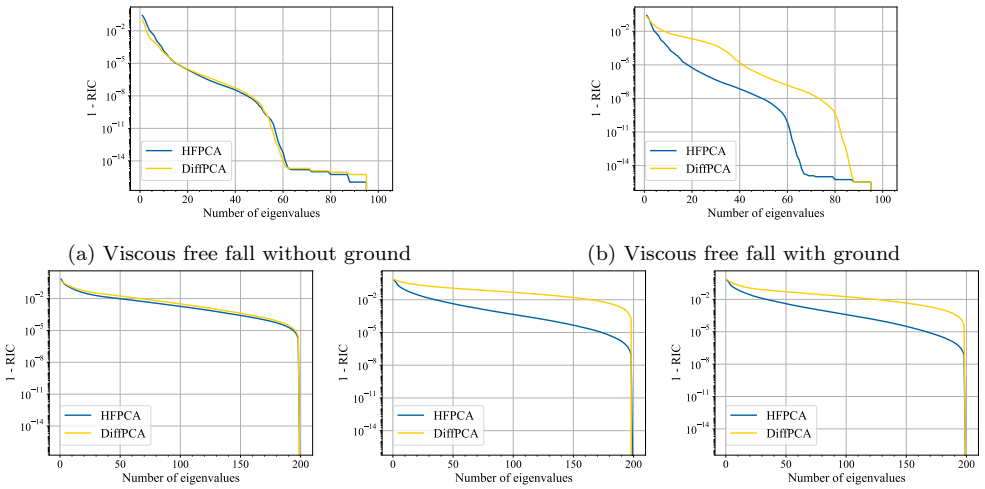
Lastly, focusing on the RAE 2822 airfoil case, Figures 20 and 22 show a notably lower ranking of the corrective surrogate for the RANS/Euler configuration compared to the RANS/RANS configuration. In this test case, DiffPCA requires significantly more principal components than HFPCA for both configurations. Consequently, the number of PCA modes needed is not the only explanation for the discrepancy in performance.

As a reminder, the prediction of the corrective approach is the low-fidelity field plus the prediction of the difference between the high- and low-fidelity simulators. In the RANS/RANS configuration, where the amplitude of the difference between the high- and low-fidelity output fields is lower than in the RANS/Euler configuration, the role of the prediction of this difference in the final high-fidelity output field prediction is limited, and vice versa in the RANS/Euler configuration. If the difference between the high- and low-fidelity output fields is poorly predicted, the RANS/Euler configuration is more affected than the RANS/RANS configuration. This can be verified by computing the relative contribution of the difference between the high- and low-fidelity output fields to the high-fidelity output field. The difference field is computed for each snapshot $i = 1, \ldots, n_v$ ($n_v$ = 1,000) of the validation set at each node $j = 1, \ldots, d_{\mathbf{y}_1}$. The value of the difference is divided by the high-fidelity value to get a relative quantity in the end. A floor is applied, such that every node whose high-fidelity value is under a threshold set to 0.05 is ignored in order to avoid divisions by numbers close to zero. Defining the set $\mathcal{J}_i = \{j \in \{1, \ldots, d_{\mathbf{y}_1}\} | y_{j,1}^{(i)} > 0.05\}$, the contribution $c_\Delta$ of the difference field to the high-fidelity field is computed as follows

$$c_\Delta = \frac{1}{n_v} \sum_{i=1}^{n_v} \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} \frac{y_{j,1}^{(i)} - y_{j,2}^{(i)}}{y_{j,1}^{(i)}}. \tag{27}$$

In the RANS/RANS configuration, $c_\Delta$ = 5.7% while it is 33.5% in the RANS/Euler configuration (about six times more).

In summary, to estimate the performance of the corrective approach for a specific test case, two points should be checked. First, as the high-fidelity, the low-fidelity, and the difference between the two fields may not be accurately




(a) Viscous free fall without ground

(b) Viscous free fall with ground

(c) NACA 0015 airfoil

(d) RAE 2822 RANS/RANS

(e) RAE 2822 RANS/Euler

Figure 28: Decay of 1 - RIC of HFPCA and DiffPCA for the various test cases, computed for 200 high- and low-fidelity samples randomly selected from the validation set

reduced by the same DR technique, practitioners should ensure that the chosen DR technique is suitable for the difference field. For instance, this can be done by computing the decay of 1 - RIC in PCA. If the DR technique is more suitable for the high-fidelity output field than for the difference field, a single-fidelity surrogate of the high-fidelity output field may outperform the corrective surrogate. Second, the relative contribution of the difference between the high- and low-fidelity output fields to the high-fidelity should be assessed. If it is small, the corrective approach is likely to yield better results than the single-fidelity surrogate of the high-fidelity. For a large relative contribution, special care should be taken to choose the DR technique so that it is appropriate for the difference field.

### *6.7.5. Manifold alignment*

In the rankings presented above (Figures 14, 16, 18, 20 and 22), the inclusion of manifold alignment consistently proves beneficial or at worst slightly detrimental in terms of prediction accuracy. The computation of manifold alignment operators entails only linear algebra operations, including a singular value decomposition in the low-dimensional latent space, which have a negligible computational overhead. Additionally, manifold alignment plays no direct role in the prediction process of high-fidelity output fields.

Hence, based on our experiments, it is advised to perform manifold alignment whenever the high- and low-fidelity manifolds are different. This is of interest when the high- and low-fidelity fields are discretized on different meshes so as to avoid a pre-processing step that projects every fidelity field onto a common mesh.

## 7. Concluding recommendations

This paper provides a comprehensive review of various techniques for constructing surrogates of a set of simulators with functional outputs. The simulators have variable fidelities and they can have different meshes for distinct fidelities. The unified framework introduced in this paper facilitates the theoretical comparison of the various existing surrogates of simulators with functional outputs, combining dimensionality reduction and latent surrogate modeling. The subsequent benchmark of a dozen of these surrogates on case studies of increasing complexity allows us to provide practical recommendations.

- Overall, multi-fidelity surrogates tend to outperform their single-fidelity counterparts when low-fidelity data is available. The lower complexity of setup of single-fidelity surrogates makes them a good starting point but multi-fidelity surrogates usually offer better predictions (see Section 6.7.1).

- Starting with linear DR is advisable. It is much simpler to train and use than nonlinear techniques, and it provides satisfactory performance (see Section 6.7.2). Depending on the decay of 1 - RIC, if too many modes must be retained to provide sufficiently accurate DR, nonlinear approaches could be tested. Special care should be taken to tuning the hyperparameters of nonlinear DR (in the reviewed methods, *e.g.*, the correlation length of the kernel of KPCA, the number of nearest neighbors in Isomap and LTSA backmapping). Note that none of the three nonlinear DR techniques reviewed in our work seem to stand out in terms of performance. Alternative nonlinear DR techniques could be investigated to improve the multi-fidelity surrogate models prediction accuracy.

- When considering fusion surrogates, MFPCA or CPCA should be preferred over LFPCA and HFPCA if the additional training and prediction computational cost is acceptable (see Section 6.7.3).

- When the high- and low-fidelity snapshots are not mapped to a common manifold, manifold alignment should be used in fusion surrogates, as it quite consistently improves the performance for a negligible additional computational cost (see Section 6.7.5).

- If the correlation between the high- and low-fidelity output fields is high, C-DiffPCA-GPR will most likely offer the best performance. When using the low-fidelity simulator to make predictions cannot be afforded, it can be replaced by a single-fidelity surrogate in the prediction process. In this case, the accuracy of this surrogate should be assessed.

- If the emulated fields have a feature moving along the mesh (*e.g.*, the boundary layer separation on the NACA 0015 airfoil described in Section 4.2), it might be difficult for linear DR to efficiently and accurately compress data. Nonlinear DR could offer better results. However, none of the reviewed surrogates using nonlinear DR showed better performances in this situation, which further encourages the investigation of alternative nonlinear DR techniques.

- Most importantly, this study shows that no surrogate is better in every case, which is another example of the No Free Lunch theorem [76]. It is therefore recommended to test several surrogates on any new task, and compare the results to opt for the most consistent solution.

**Acknowledgment**

This work is co-funded by ONERA and the Agence de l'innovation de défense.

**Reproducibility and benchmark data**

The python code used for the different multi-fidelity surrogate models, along with the training and validation datasets, are available upon request.

**Appendix A. Numerical settings**

- **Hardware:** all repetitions are run on an Intel® Xeon® E5-2650 v4 (2.20GHz) CPU

- **Dimensionality reduction**
  - **PCA:** the Singular Value Decomposition (SVD) of `scipy` [67] is used, {∅, SF, HF, LF, MF, Diff, G}PCA are truncated with a RIC of 99.9%.
  - **CPCA:** all modes are retained in the SVDs at all stages.
  - **Kernel PCA:** the implementation of `scikit-learn` [77] is used, the size of the latent space is set to $n_1 - 1$, the kernel function is RBF. The inverse mapping is performed with the approach described in [78]. Both the hyperparameters of the forward and inverse mapping are computed by minimizing the DR error with a $k$-fold cross-validation strategy with $k = \lfloor 0.5 \times n_2 \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function.
  - **Isomap and LTSA:** the implementation of `scikit-learn` [77] is used for the mapping from the high-dimensional space to the latent space. The training of the hyperparameters and the inverse mapping is done as explained by Decker *et al.* [6], *i.e.*, the dimensionality of the latent space is set to the dimensionality of the input variable space $\dim(U)$, the number of nearest neighbors is computed by minimizing the Kruskal's stress [59, 79] in Isomap and the variance of distance ratios [60] in LTSA and the inverse mapping is the one described by Roweis and Saul [80].
- **Intermediate surrogate modeling**
  - **Single-fidelity GP regression:** single-fidelity GP regression is performed with `smt` [72] with constant trend. The kernel function is Matérn-5/2 whose hyperparameters are computed by maximizing the likelihood with the COBYLA algorithm [81] with 20 restarts, the default nugget value is used.
  - **GP regression with mixed continuous and discrete variables:** the implementation of `smt` [72] is used, the kernel function for continuous variables is Matérn-5/2 and the kernel function of discrete variables is based on Gower distance [82]. The hyperparameters are computed by maximizing the likelihood with the COBYLA algorithm [81] with 20 restarts, the default nugget value is used.

  - **Multi-fidelity AR1 co-Kriging:** a modified version of the AR1 co-Kriging of `OpenMDAO` [83] that provides support for multistart is used. The regression function is constant and the kernel function is Matérn-5/2 whose hyperparameters are computed by maximizing the likelihood with the COBYLA algorithm [81] with 20 restarts. The nugget value is set to $10^{-10}$.
  - **GP regression with tensorized covariance:** it has been implemented following the description of Kerleguer [3], with a Matérn-5/2 kernel function with no nugget value. The hyperparameters are computed by minimizing the prediction error with a $k$-fold cross-validation strategy with $k = \min(10, n_1)$ with the COBYLA algorithm [81].

- **Design of experiments**
  - **Nested LHS:** a modified version of the implementation of `smt` [72] is used, adding the support for variable $n_1/n_2$ ratios.

## Appendix B. Decomposition of the normed RMSE into normed DR and intermediate surrogate modeling RMSE

Tables B.3 to B.5 show that for the viscous free fall and NACA 0015 test cases, the main contribution to the prediction normed RMSE is the error due to the intermediate surrogate. For the two RAE 2822 test cases (Tables B.6 and B.7), the breakdown of the prediction RMSE is more even, although the intermediate surrogate remains the main contributor.

Table B.3: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for the viscous free fall without ground test case, for every configuration of $(n_1, n_2)$ combined.

| surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|
| F-HFPCA-AR1 | 8.3 | 0.2 | 8.3 |
| F-LFPCA-AR1 | 9.7 | 3.7 | 8.1 |
| F-MFPCA-AR1 | 8.2 | 0.1 | 8.2 |
| F-MFPCA-CategGPR | 9.0 | 0.1 | 9.0 |
| F-SFPCA-AR1 | 8.3 | 0.2 | 8.3 |
| F-SFPCA-MA-AR1 | 8.4 | 0.2 | 8.4 |
| S-HFPCA-GPR | 18.3 | 0.2 | 18.3 |

Table B.4: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for the viscous free fall with ground test case, for every configuration of $(n_1, n_2)$ combined.

| surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|
| F-HFPCA-AR1 | 9.3 | 0.2 | 9.3 |
| F-LFPCA-AR1 | 10.3 | 3.1 | 9.4 |
| F-MFPCA-AR1 | 9.4 | 0.1 | 9.4 |
| F-MFPCA-CategGPR | 11.9 | 0.1 | 11.9 |
| F-SFPCA-AR1 | 9.5 | 0.1 | 9.5 |
| F-SFPCA-MA-AR1 | 9.3 | 0.1 | 9.3 |
| S-HFPCA-GPR | 18.3 | 0.2 | 18.3 |

Table B.5: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for the NACA 0015 test case, for every configuration of $(n_1, n_2)$ combined.

| surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|
| F-HFPCA-AR1 | 19.6 | 5.4 | 19.2 |
| F-LFPCA-AR1 | 22.0 | 12.9 | 19.7 |
| F-MFPCA-AR1 | 9.4 | 0.1 | 9.4 |
| F-MFPCA-CategGPR | 19.1 | 2.5 | 19.1 |
| F-SFPCA-AR1 | 20.0 | 5.3 | 19.7 |
| F-SFPCA-MA-AR1 | 20.0 | 5.3 | 19.6 |
| S-HFPCA-GPR | 20.6 | 5.4 | 20.2 |

Table B.6: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for the RANS/RANS configuration of the RAE 2822 airfoil test case, for every configuration of $(n_1, n_2)$ combined.

| surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|
| F-HFPCA-AR1 | 22.2 | 14.7 | 17.1 |
| F-LFPCA-AR1 | 22.3 | 11.7 | 19.5 |
| F-MFPCA-AR1 | 21.7 | 7.9 | 20.0 |
| F-MFPCA-CategGPR | 23.1 | 7.9 | 21.6 |
| F-SFPCA-AR1 | 24.0 | 14.6 | 20.2 |
| F-SFPCA-MA-AR1 | 22.2 | 14.6 | 16.8 |
| S-HFPCA-GPR | 26.9 | 14.7 | 25.2 |

Table B.7: Decomposition of the normed RMSE $e^{\text{norm}}$ into normed DR RMSE $e_{\text{dr}}^{\text{norm}}$ and normed intermediate surrogate modeling RMSE $e_{\text{ism}}^{\text{norm}}$ for the RANS/Euler configuration of the RAE 2822 airfoil test case, for every configuration of $(n_1, n_2)$ combined.

| surrogate | $e^{\text{norm}}$ (%) | $e_{\text{dr}}^{\text{norm}}$ (%) | $e_{\text{ism}}^{\text{norm}}$ (%) |
|---|---|---|---|
| F-HFPCA-AR1 | 25.2 | 14.7 | 22.1 |
| F-LFPCA-AR1 | 30.4 | 20.0 | 23.5 |
| F-MFPCA-AR1 | 27.7 | 7.9 | 26.4 |
| F-MFPCA-CategGPR | 28.6 | 7.9 | 27.5 |
| F-SFPCA-AR1 | 25.6 | 14.6 | 22.0 |
| F-SFPCA-MA-AR1 | 24.7 | 14.6 | 21.2 |
| S-HFPCA-GPR | 27.2 | 14.7 | 25.5 |

## Appendix C. Comparison of the decay of 1 - RIC for {HF, LF, MF, Diff}PCA

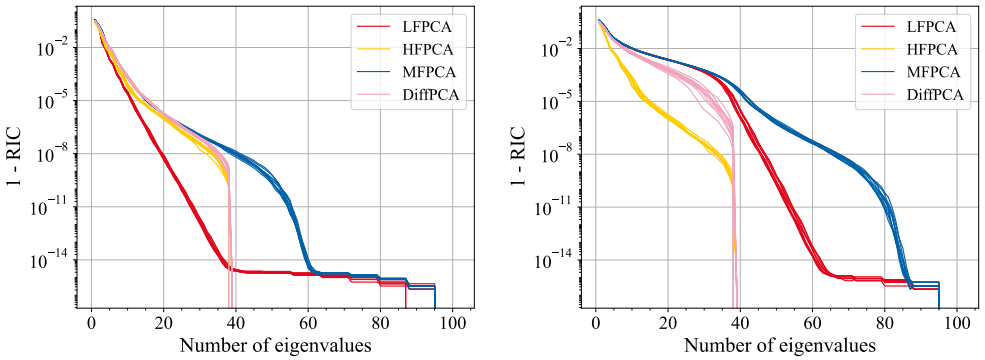



Figure C.29: Decay of 1 - RIC for the different PCAs for the viscous free fall without ground (left) and with a ground (right) test cases with $n_1 = 40$ and $n_2 = 400$ for 10 repetitions each.

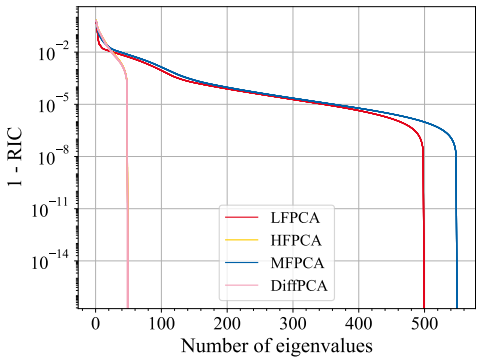



Figure C.30: Decay of 1 - RIC for the different PCAs for the NACA 0015 test case with $n_1 = 50$ and $n_2 = 500$ for 10 repetitions each.

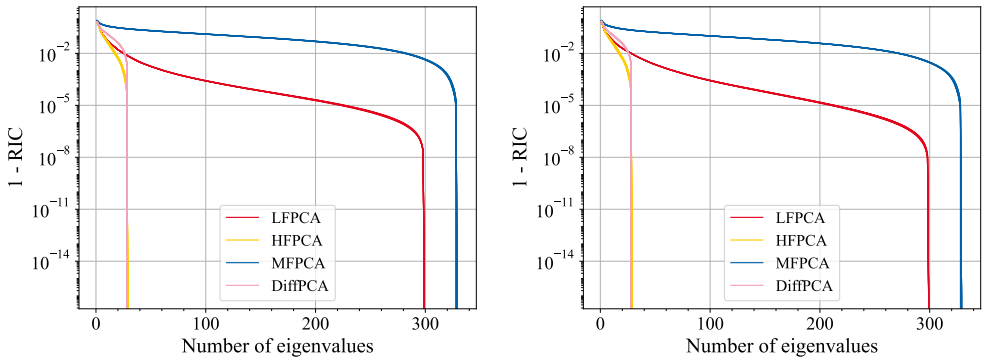



Figure C.31: Decay of 1 - RIC for the different PCAs for the RAE 2822 airfoil test case RANS/RANS configuration (left) and RANS/Euler configuration (right) with $n_1 = 30$ and $n_2 = 300$ for 10 repetitions each.

## Appendix D. Ten samples of the difference between the high- and low-fidelity output fields for the various test cases

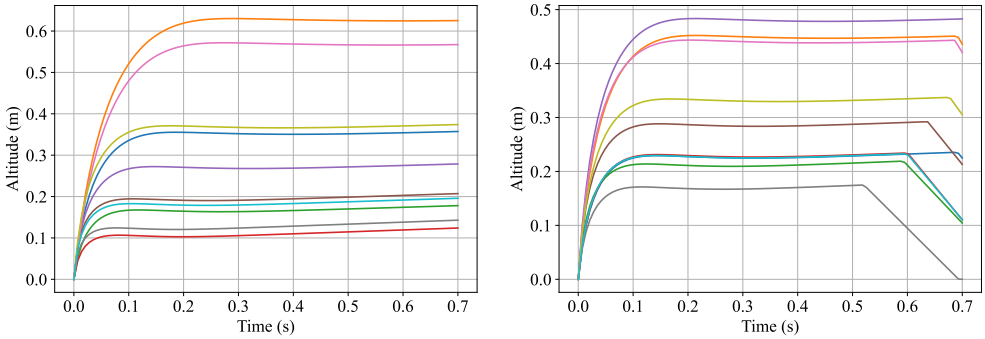



Figure D.32: Ten samples of the difference between the high- and low-fidelity output fields for the viscous free fall without a ground (left) and with a ground (right) test cases.

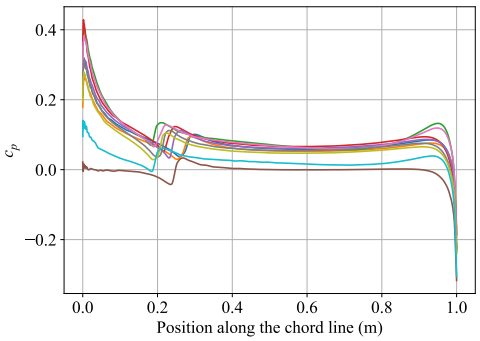



Figure D.33: Ten samples of the difference between the high- and low-fidelity output fields for the NACA 0015 test case.

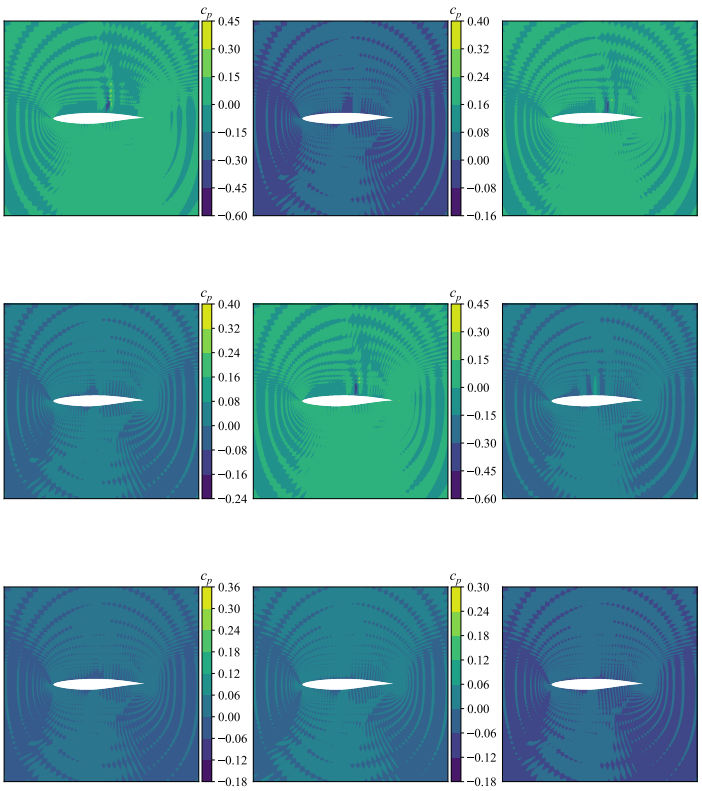



Figure D.34: Nine samples of the difference between the high- and low-fidelity output fields for the RAE 2822 RANS/RANS test case.

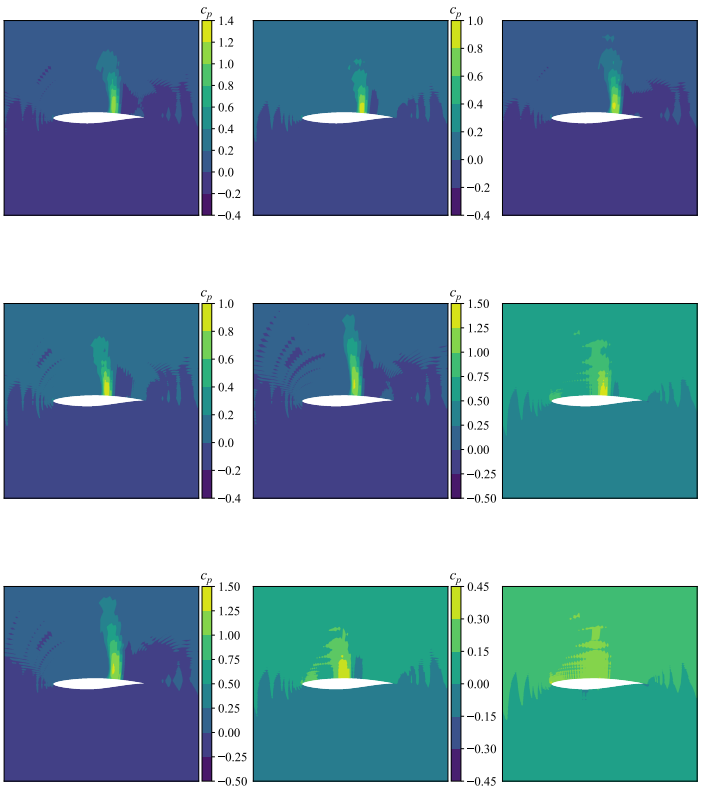



Figure D.35: Nine samples of the difference between the high- and low-fidelity output fields for the RAE 2822 RANS/Euler test case.

## References

[1] T. Benamara, P. Breitkopf, I. Lepot, C. Sainvitu, P. Villon, Multi-fidelity POD surrogate-assisted optimization: concept and aero-design study, Structural and Multidisciplinary Optimization 56 (6) (2017) 1387–1412. doi:10.1007/s00158-017-1730-4.

[2] S. Bunnell, S. Gorrell, J. Salmon, Multi-fidelity surrogates from shared principal components, Structural and Multidisciplinary Optimization 63 (5) (2021) 2177–2190. doi:10.1007/s00158-020-02793-z.

[3] B. Kerleguer, Multifidelity surrogate modeling for time-series outputs,

SIAM/ASA Journal on Uncertainty Quantification 11 (2) (2023) 514–539. doi:10.1137/20M1386694.

[4] B. Malouin, J.-Y. Trépanier, M. Gariepy, Interpolation of transonic flows using a proper orthogonal decomposition method, International Journal of Aerospace Engineering 2013 (Jan. 2013). doi:10.1155/2013/928904.

[5] C. Perron, D. Rajaram, D. N. Mavris, Multi-fidelity non-intrusive reduced-order modelling based on manifold alignment, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 477 (2253) (2021) 20210495. doi:10.1098/rspa.2021.0495.

[6] K. Decker, N. Iyengar, D. Rajaram, C. Perron, D. Mavris, Manifold alignment-based nonintrusive and nonlinear multifidelity reduced-order modeling, AIAA Journal 61 (1) (2022) 454–474. doi:10.2514/1.J061720.

[7] T. Rokita, P. P. Friedmann, Multifidelity coKriging for high-dimensional output functions with application to hypersonic airloads computation, AIAA Journal 56 (8) (2018) 3060–3070. doi:10.2514/1.J056620.

[8] A. Thenon, V. Gervais, M. L. Ravalec, Multi-fidelity meta-modeling for reservoir engineering - Application to history matching, Computational Geosciences 20 (6) (2016) 1231–1250. doi:10.1007/s10596-016-9587-y.

[9] D. J. J. Toal, On the potential of a multi-fidelity G-POD based approach for optimization and uncertainty quantification, in: ASME Turbo Expo 2014: Turbine Technical Conference and Exposition, American Society of Mechanical Engineers Digital Collection, 2014. doi:10.1115/GT2014-25184.

[10] X. Wang, J. Kou, W. Zhang, Multi-fidelity surrogate reduced-order modeling of steady flow estimation, International Journal for Numerical Methods in Fluids 92 (12) (2020) 1826–1844. doi:10.1002/fld.4850.

[11] M. J. Mifsud, D. G. MacManus, S. Shaw, A variable-fidelity aerodynamic model using proper orthogonal decomposition, International Journal for Numerical Methods in Fluids 82 (10) (2016) 646–663. doi:10.1002/fld.4234.

[12] L. Parussini, D. Venturi, P. Perdikaris, G. E. Karniadakis, Multi-fidelity Gaussian process regression for prediction of random fields, Journal of Computational Physics 336 (2017) 36–50. doi:10.1016/j.jcp.2017.01.047.

[13] M. Guo, A. Manzoni, M. Amendt, P. Conti, J. S. Hesthaven, Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114378. doi:10.1016/j.cma.2021.114378.

[14] S. Li, Z. Wang, R. Kirby, S. Zhe, Deep multi-fidelity active learning of high-dimensional outputs, in: Proceedings of The 25th International Conference on Artificial Intelligence and Statistics, PMLR, 2022, pp. 1694–1711.

[15] H. Yang, J. Ou, Y. Wang, Neural-physics multi-fidelity model with active learning and uncertainty quantification for GPU-enabled microfluidic concentration gradient generator design, Computer Methods in Applied Mechanics and Engineering 417 (2023) 116434. doi:10.1016/j.cma.2023.116434.

[16] S.-Y. Hsu, C.-L. Chang, J. Samareh, A simplified mesh deformation method using commercial structural analysis software, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, American Institute of Aeronautics and Astronautics, 2004. doi:10.2514/6.2004-4409.

[17] F. Camastra, A. Staiano, Intrinsic dimension estimation: advances and open problems, Information Sciences 328 (2016) 26–41. doi:10.1016/j.ins.2015.08.029.

[18] T. J. Santner, B. J. Williams, W. I. Notz, Space-filling designs for computer experiments, in: T. J. Santner, B. J. Williams, W. I. Notz (Eds.), The Design and Analysis of Computer Experiments, Springer Series in Statistics, Springer, New York, NY, 2018, pp. 145–200. doi:10.1007/978-1-4939-8847-1_5.

[19] L. van der Maaten, E. Postma, J. van den Herik, Dimensionality reduction: a comparative review, TiCC TR 2009–005 (2009) 36.

[20] W. Jia, M. Sun, J. Lian, S. Hou, Feature dimensionality reduction: a review, Complex & Intelligent Systems 8 (3) (2022) 2663–2693. doi:10.1007/s40747-021-00637-x.

[21] X. Huang, L. Wu, Y. Ye, A review on dimensionality reduction techniques, International Journal of Pattern Recognition and Artificial Intelligence (Sep. 2019). doi:10.1142/S0218001419500174.

[22] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, C. G. Wu, Proper orthogonal decomposition and its applications — Part I: theory, Journal of Sound and Vibration 252 (3) (2002) 527–544. doi:10.1006/jsvi.2001.4041.

[23] J. Mercer, A. R. Forsyth, Functions of positive and negative type, and their connection the theory of integral equations, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character 209 (441-458) (1909) 415–446. doi:10.1098/rsta.1909.0016.

[24] R. Pinnau, Model reduction via proper orthogonal decomposition, in: W. H. A. Schilders, H. A. van der Vorst, J. Rommes (Eds.), Model Order Reduction: Theory, Research Aspects and Applications, Mathematics in Industry, Springer, Berlin, Heidelberg, 2008, pp. 95–109. doi:10.1007/978-3-540-78841-6_5.

[25] R. L. Gorsuch, Basic factor models, in: Factor Analysis, 2nd Edition, Routledge, 2014.

[26] P. Comon, Independent component analysis, a new concept?, Signal Processing 36 (3) (1994) 287–314. doi:10.1016/0165-1684(94)90029-9.

[27] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319. doi:10.1162/089976698300017467.

[28] B. Schölkopf, The kernel trick for distances, in: Advances in Neural Information Processing Systems, Vol. 13, MIT Press, 2000.

[29] J. Tenenbaum, Mapping a manifold of perceptual observations, in: Advances in Neural Information Processing Systems, Vol. 10, MIT Press, 1997.

[30] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, SIAM Journal on Scientific Computing 26 (1) (2004) 313–338. doi:10.1137/S1064827502419154.

[31] D. Bank, N. Koenigstein, R. Giryes, Autoencoders, in: L. Rokach, O. Maimon, E. Shmueli (Eds.), Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook, Springer International Publishing, Cham, 2023, pp. 353–374. doi:10.1007/978-3-031-24628-9_16.

[32] J. Ham, D. D. Lee, L. Saul, Learning high dimensional correspondences from low dimensional manifolds, in: 20th International Conference on Machine Learning (ICML 2003) Workshop: TheContinuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, Washington D.C., 2003.

[33] C. Wang, S. Mahadevan, Manifold alignment without correspondence, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009, pp. 1273–1278.

[34] C. Wang, S. Mahadevan, Manifold alignment using Procrustes analysis, in: Proceedings of the 25th international conference on Machine learning, ICML '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1120–1127. doi:10.1145/1390156.1390297.

[35] J. C. Gower, Procrustes methods, WIREs Computational Statistics 2 (4) (2010) 503–508. doi:10.1002/wics.107.

[36] J. C. Gower, G. B. Dijksterhuis, Procrustes problems, Vol. 30, Oxford University Press, Oxford, UK, 2004.

[37] J. Rougier, Efficient emulators for multivariate deterministic functions, Journal of Computational and Graphical Statistics 17 (4) (2008) 827–843. doi:10.1198/106186008X384032.

[38] G. Perrin, Adaptive calibration of a computer code with time-series output, Reliability Engineering & System Safety 196 (2020) 106728. doi:10.1016/j.ress.2019.106728.

[39] H. Khatouri, T. Benamara, P. Breitkopf, J. Demange, Metamodeling techniques for CPU-intensive simulation-based design optimization: a survey, Advanced Modeling and Simulation in Engineering Sciences 9 (1) (2022) 1. doi:10.1186/s40323-022-00214-y.

[40] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Review 60 (3) (2018) 550–591. doi:10.1137/16M1082469.

[41] M. G. Fernández-Godino, Review of multi-fidelity models (Apr. 2023). arXiv:1609.07196, doi:10.48550/arXiv.1609.07196.

[42] G. Matheron, Principles of geostatistics, Economic Geology 58 (8) (1963) 1246–1266. doi:10.2113/gsecongeo.58.8.1246.

[43] M. Eldred, Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design, in: 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, 2009. doi:10.2514/6.2009-2274.

[44] D. F. Specht, A general regression neural network, IEEE transactions on neural networks 2 (6) (1991) 568–576. doi:10.1109/72.97934.

[45] A. Cutler, D. R. Cutler, J. R. Stevens, Random forests, in: C. Zhang, Y. Ma (Eds.), Ensemble Machine Learning: Methods and Applications, Springer, New York, NY, 2012, pp. 157–175. doi:10.1007/978-1-4419-9326-7_5.

[46] M. D. Buhmann, Radial basis functions, Acta Numerica 9 (2000) 1–38. doi:10.1017/S0962492900000015.

[47] Z.-H. Han, S. Görtz, Hierarchical kriging model for variable-fidelity surrogate modeling, AIAA Journal 50 (9) (2012) 1885–1896. doi:10.2514/1.J051354.

[48] L. Le Gratiet, J. Garnier, Recursive co-Kriging model for design of computer experiments with multiple levels of fidelity, International Journal for Uncertainty Quantification 4 (5) (2014) 365–386. doi:10.1615/Int.J.UncertaintyQuantification.2014006914.

[49] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, G. E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 473 (2198) (2017) 20160751. doi:10.1098/rspa.2016.0751.

[50] A. Damianou, N. D. Lawrence, Deep gaussian processes, in: Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, PMLR, 2013, pp. 207–215.

[51] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, J. González, Deep gaussian processes for multi-fidelity modeling, arXiv, 2019. arXiv:1903.07320, doi:10.48550/arXiv.1903.07320.

[52] L. W.-T. Ng, M. Eldred, Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-1852.

[53] C. Durantin, J. Rouxel, J.-A. Désidéri, A. Glière, Multifidelity surrogate modeling based on radial basis functions, Structural and Multidisciplinary Optimization 56 (5) (2017) 1061–1075. doi:10.1007/s00158-017-1703-7.

[54] H. Liu, J. Cai, Y.-S. Ong, Remarks on multi-output Gaussian process regression, Knowledge-Based Systems 144 (2018) 102–121. doi:10.1016/j.knosys.2017.12.034.

[55] D. J. J. Toal, Applications of multi-fidelity multi-output Kriging to engineering design optimization, Structural and Multidisciplinary Optimization 66 (6) (2023) 125. doi:10.1007/s00158-023-03567-z.

[56] D. Pigoli, A. Menafoglio, P. Secchi, Kriging prediction for manifold-valued random fields, Journal of Multivariate Analysis 145 (2016) 117–131. doi:10.1016/j.jmva.2015.12.006.

[57] R. Everson, L. Sirovich, Karhunen–Loève procedure for gappy data, JOSA A 12 (8) (1995) 1657–1664. doi:10.1364/JOSAA.12.001657.

[58] W. Cardwell, Jr., R. Parsons, Average permeabilities of heterogeneous oil sands, Transactions of the AIME 160 (01) (1945) 34–42. doi:10.2118/945034-G.

[59] J. B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1) (1964) 1–27. doi:10.1007/BF02289565.

[60] L. Shi, Q. Yang, Y. Xu, P. He, A model of selecting the parameters based on the variance of distance ratios for manifold learning algorithms, in: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 2, 2009, pp. 507–512. doi:10.1109/FSKD.2009.471.

[61] C. E. Rasmussen, Gaussian processes for machine learning, in: Advanced lectures on machine learning, Vol. 1, Spinger, 2006, pp. 63–71.

[62] A viscous free fall example — OpenTURNS 1.21 documentation, https://openturns.github.io/openturns/latest/usecases/use_case_viscous_fall.html.

[63] P. Timmerman, K. Weele, On the rise and fall of a ball with linear or quadratic drag, American Journal of Physics - AMER J PHYS 67 (1999) 538–546. doi:10.1119/1.19320.

[64] L. Petzold, Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations, SIAM Journal on Scientific and Statistical Computing 4 (1) (1983) 136–148. doi:10.1137/0904010.

[65] P. B. C. Leal, AeroPy: alpha version, Zenodo (2016). doi:10.5281/zenodo.45993.

[66] M. Drela, XFOIL: an analysis and design system for low reynolds number airfoils, in: T. J. Mueller (Ed.), Low Reynolds Number Aerodynamics, Lecture Notes in Engineering, Springer, Berlin, Heidelberg, 1989, pp. 1–12. doi:10.1007/978-3-642-84010-4_1.

[67] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H.

Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nature Methods 17 (3) (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[68] G. Kenway, G. Kennedy, J. R. R. A. Martins, A CAD-free approach to high-fidelity aerostructural optimization, in: 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2010-9231.

[69] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, J. J. Alonso, SU2: an open-source suite for multiphysics simulation and design, AIAA Journal 54 (3) (2016) 828–846. doi:10.2514/1.J053813.

[70] R. P. Dwight, Robust mesh deformation using the linear elasticity equations, in: H. Deconinck, E. Dick (Eds.), Computational Fluid Dynamics 2006, Springer, Berlin, Heidelberg, 2009, pp. 401–406. doi:10.1007/978-3-540-92779-2_62.

[71] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 1992. doi:10.2514/6.1992-439.

[72] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, J. R. R. A. Martins, SMT 2.0: a surrogate modeling toolbox with a focus on hierarchical and mixed variables Gaussian processes (Aug. 2023). arXiv:2305.13998, doi:10.48550/arXiv.2305.13998.

[73] H. Kang, Z. Tian, G. Chen, L. Li, T. Chu, Investigation on the nonintrusive multi-fidelity reduced-order modeling for PWR rod bundles, Nuclear Engineering and Technology 54 (5) (2022) 1825–1834. doi:10.1016/j.net.2021.10.036.

[74] C. Perron, Multi-fidelity reduced-order modeling applied to fields with inconsistent representations, Ph.D. thesis, Georgia Institute of Technology (Dec. 2020).

[75] N. Lüthen, S. Marelli, B. Sudret, Sparse polynomial chaos expansions: literature survey and benchmark, SIAM/ASA Journal on Uncertainty Quantification 9 (2) (2021) 593–649. doi:10.1137/20M1315774.

[76] Y. Ho, D. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, Journal of Optimization Theory and Applications 115 (3) (2002) 549–570. doi:10.1023/A:1021251113462.

[77] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: machine learning in python, Journal of Machine Learning Research 12 (85) (2011) 2825–2830.

[78] G. H. Bakır, J. Weston, B. Schölkopf, Learning to find pre-images, in: Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03, MIT Press, Cambridge, MA, USA, 2003, pp. 449–456.

[79] T. Franz, R. Zimmermann, S. Görtz, N. Karcher, Interpolation-based reduced-order modelling for steady transonic flows via manifold learning, International Journal of Computational Fluid Dynamics 28 (3-4) (2014) 106–121. doi:10.1080/10618562.2014.918695.

[80] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326. doi:10.1126/science.290.5500.2323.

[81] M. J. D. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, in: S. Gomez, J.-P. Hennart (Eds.), Advances in Optimization and Numerical Analysis, Mathematics and Its Applications, Springer Netherlands, Dordrecht, 1994, pp. 51–67. doi:10.1007/978-94-015-8330-5_4.

[82] P. Saves, Y. Diouane, N. Bartoli, T. Lefebvre, J. Morlier, A general square exponential kernel to handle mixed-categorical variables for Gaussian process, in: AIAA AVIATION 2022 Forum, American Institute of Aeronautics and Astronautics, 2022. doi:10.2514/6.2022-3870.

[83] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, B. A. Naylor, OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization, Structural and Multidisciplinary Optimization 59 (4) (2019) 1075–1104. doi:10.1007/s00158-019-02211-z.