International Journal for Uncertainty Quantification, 12(4):39–64 (2022)

MANIFOLD LEARNING-BASED POLYNOMIAL CHAOS EXPANSIONS FOR HIGH-DIMENSIONAL SURROGATE MODELS

Katiana Kontolati,[1] Dimitrios Loukrezis,[2,3] Ketson R.M. dos Santos,[4] Dimitrios G. Giovanis,[1] & Michael D. Shields[1,*]

[1]*Department of Civil & Systems Engineering, Johns Hopkins University, Baltimore, Maryland, USA*

[2]*Institute for Accelerator Science and Electromagnetic Fields, Technische Universität Darmstadt, Darmstadt, Germany*

[3]*Centre for Computational Engineering, Technische Universität Darmstadt, Darmstadt, Germany*

[4]*Earthquake Engineering and Structural Dynamics Laboratory, École Polytechnique Fédérale de Lausanne, VD, Switzerland*

*Address all correspondence to: Michael D. Shields, Department of Civil & Systems Engineering, Johns Hopkins University, Baltimore, Maryland, USA, E-mail: michael.shields@jhu.edu

Original Manuscript Submitted: 7/20/2021; Final Draft Received: 1/9/2022

In this work we introduce a manifold learning-based method for uncertainty quantification (UQ) in systems describing complex spatiotemporal processes. Our first objective is to identify the embedding of a set of high-dimensional data representing quantities of interest of the computational or analytical model. For this purpose, we employ Grassmannian diffusion maps, a two-step nonlinear dimension reduction technique which allows us to reduce the dimensionality of the data and identify meaningful geometric descriptions in a parsimonious and inexpensive manner. Polynomial chaos expansion is then used to construct a mapping between the stochastic input parameters and the diffusion coordinates of the reduced space. An adaptive clustering technique is proposed to identify an optimal number of clusters of points in the latent space. The similarity of points allows us to construct a number of geometric harmonic emulators which are finally utilized as a set of inexpensive pretrained models to perform an inverse map of realizations of latent features to the ambient space and thus perform accurate out-of-sample predictions. Thus, the proposed method acts as an encoder-decoder system which is able to automatically handle very high-dimensional data while simultaneously operating successfully in the small-data regime. The method is demonstrated on two benchmark problems and on a system of advection-diffusion-reaction equations which model a first-order chemical reaction between two species. In all test cases, the proposed method is able to achieve highly accurate approximations which ultimately lead to the significant acceleration of UQ tasks.

KEY WORDS: *surrogate modeling, manifold learning, low-dimensional embedding, large-scale computational systems, Grassmann manifold, uncertainty quantification, advection-diffusion-reaction*

1. INTRODUCTION

Robust engineering design and optimal decision making require the accurate prediction of the performance of (often complex) stochastic systems or systems with significant uncertainty. Uncertainty quantification (UQ), an important field of computational science and engineering, provides a means of propagating uncertainties through the system to understand their influence on responses of interest. Despite the recent progress in hardware and processing power, UQ is often prohibitively expensive for real-world systems of interest, as it usually requires a large number of evaluations of complex mathematical models. To alleviate this issue, surrogate models are employed to establish an efficient

approximate mapping between model inputs and outputs. Such models enable the propagation of mixed aleatoric and epistemic uncertainties across scales [1]. The construction of accurate surrogates, however, typically requires smooth input-output functional relations, which may not be realistic in real-world applications, that predict low-dimensional quantities of interest that may not reflect the complexity of the solution.

Nonintrusive polynomial chaos expansions (PCE) are an established and versatile surrogate modeling technique that express model input-output relations in terms of an expansion of polynomials that are orthonormal with respect to the probability density function (PDF) characterizing the input random variables [1–5]. One of the main advantages of PCE methods is that UQ tasks such as moment estimation and sensitivity analysis can be easily performed by post-processing the terms of the PCE [6–8]. In the case of a high-dimensional input parameter space, sparse PCE methods have been successfully proposed in the literature, which take advantage of the so-called "sparsity of effects" principle to construct surrogates with only a small number of forward model evaluations [9–11]. In such cases, however, a suitable model selection criterion must be employed for tuning the hyperparameters that are used to obtain the optimal model. Several techniques exist to adaptively identify the optimal polynomial basis and associated sparse solution, which aim to keep the size of the basis small by controlling which functions are added to the basis [10,12–14]. Despite its advantages, in cases where very high-dimensional outputs are considered, a PCE surrogate can be both computationally intractable to construct and incapable of accurately performing out-of-sample predictions.

One way to overcome the challenges associated with high-dimensional models is to apply dimension reduction techniques. Linear and nonlinear dimension reduction methods can be used to map data onto lower-dimensional manifolds (embeddings) by identifying and extracting meaningful features. Such techniques are important to overcome the so-called "curse of dimensionality," to avoid overfitting, to denoise data, and to enable regression analysis tasks. Although dimension reduction methods were originally developed for computer vision and image recognition applications, they have been increasingly used in recent years to facilitate the construction of accurate surrogates for high-dimensional physics-based models.

Several methods use linear spectral decomposition methods such as principal component analysis (PCA), which involve the eigendecomposition of the data covariance function to capture the dominant modes of the output represented in the form of a field (matrix) [15,16]. Similarly, proper orthogonal decomposition (POD) or the Karhunen-Loève expansion (KLE) has been widely used for reduced-order model construction [17–20]. Active subspaces, a dimension reduction technique which discovers linear manifolds of the data, has been proposed as an in-built technique for the construction of Gaussian process (GP) surrogates [21–24]. Furthermore, multiple gradient-based techniques can be found in the literature for identifying subspaces in situations involving multivariate outputs and high-dimensional input parameter spaces [18,19,25–27].

Nonlinear dimension reduction, also known as manifold learning, is used to deal with the limitations of the linear methods, namely the assumption that high-dimensional data can be embedded in linear spaces. Instead, nonlinear dimension reduction methods consider that the data reside on some low-dimensional, nonlinear manifold such as a Grassmannian or a diffusion manifold. Recent work of the co-authors has considered the construction of surrogate models on the Grassmannian [28–30]. Another class of methods leverages diffusion maps (DMaps) [31], to either draw samples from a distribution on the diffusion manifold [32–34] or construct surrogate models on the diffusion manifold [35,36]. Additionally, in a recent work kernel PCA is coupled with kriging and PCE to extend surrogates to high-dimensional models [37].

An alternative approach to identify latent representations of data is to use deep neural networks (DNNs) such as multilayer perceptrons (MLPs) [38,39]. Recently, multiple techniques based on autoencoders (unsupervised learning) [40] and convolutional neural networks (supervised learning) [41], have been proposed for constructing surrogate models when input and output fields are high-dimensional [42–49]. Such methods have lately received increasing attention primarily due to advancements in computer hardware and the availability of powerful resources such as graphical and tensor processing units (GPUs, TPUs). Although DNNs are capable of capturing complex nonlinear relations between high-dimensional inputs and outputs and provide both encoder and decoder paths for dimension reduction purposes, they are still considered more suitable for problems in the so-called "big-data" regime. Furthermore, such models are very costly to train, and rely on the heuristic choice of the network architecture and the calibration of multiple hyperparameters. Finally, they do not inherently provide a link between the input stochastic parameters and model output, which is essential for the implementation of UQ tasks.

In this work we introduce a novel framework, which combines low-dimensional manifold learning principles with surrogate model construction for the interpolation of dimension-reduced solutions that can be employed to generate out-of-sample predictions based on a limited number of model evaluations. We are interested in complex models that generate high-dimensional outputs (e.g., high-fidelity finite element models) that are computationally expensive to run. As a result, we can only afford a small number of model evaluations. Dimensionality reduction is achieved using the Grassmannian diffusion maps (GDMaps) technique introduced in [50], which identifies a latent representation of the dataset on a lower-dimensional manifold via a two-step procedure. In the first step, high-dimensional data (model solutions) are projected onto an orthonormal matrix manifold called the Grassmann manifold [51,52] that defines the subspace structure of the data. In the second step the diffusion maps (DMaps) method is employed to unfold the underlying nonlinear geometry of the data on the Grassmann manifold onto a diffusion manifold. Next, a PCE surrogate model is constructed to establish a mapping between input parameters and coordinates on the diffusion manifold. To reconstruct the full solution from PCE prediction solutions on the diffusion, a set of special functions called geometric harmonics (GH) [53] is used to locally define suitable mappings from the data on the diffusion manifold onto the tangent space of the Grassmann manifold. The local GH models allow us to perform out-of-sample predictions and return generated points on the diffusion manifold to the physically interpretable space.

The advantages of the proposed method lie in its ability to automatically handle high-dimensional datasets generated by complex models and extract important low-dimensional descriptors which sufficiently represent the complex physics of the system. Furthermore, our approach enables the minimization of necessary model simulations as it works well in the small-data regime, greatly reduces training time, and provides a direct way to decode the compressed data to the original space and link input parameters with model outputs. We show that the proposed method is robust and allows the acceleration of UQ tasks in cases of nonlinear complex applications.

The rest of this paper is organized as follows. The theoretical background for the dimension reduction methods, geometric harmonics, and PCE surrogates employed in this work, is briefly presented in Section 2. The important ingredients of the proposed framework are discussed in detail in Section 3. This is divided into two sections where we describe an "encoder path" that follows the Grassmannian diffusion maps and PCE surrogates on the manifold, and a "decoder path" that describes the construction of local geometric harmonics to generate full solutions from reduced order predictions. The performance of the proposed approach is assessed by three illustrative applications given in Section 5. The first example involves a model problem from electromagnetic field theory. In the second example, the method is applied to predict time evolution on the classic Lotka-Volterra (predator-prey) dynamical system. The third application deals with a system of advection-diffusion-reaction equations modeling a first-order chemical reaction between two species. Finally, Section 6 presents the conclusions.

The distinct components of the proposed method (GDMaps and PCE surrogate modeling) have been individually implemented in UQpy (Uncertainty Quantification with python) a general-purpose open-source software for modeling uncertainty in physical and mathematical systems [54]. Codes for implementing the proposed framework and reproducing the results are available at https://github.com/katiana22/GDM-PCE.

2. PRELIMINARIES

2.1 Grassmannian Diffusion Maps (GDMaps)

Diffusion maps (DMaps) [31] is a manifold learning technique that is based on the construction of a Markov transition probability matrix corresponding to a random walk on a graph connecting the data. The vertices of the graph are the data points and the edges represent connections between the data points that are weighted by transition probabilities representing the local similarities between pairs of points. The graph structure can be parameterized by the so-called diffusion coordinates, representing the low-dimensional manifold (embedding) of the data. To identify this parameterization, a careful selection (ideally parsimonious [55]) of the eigenvectors of the Markov matrix needs to be performed.

Grassmannian diffusion maps (GDMaps) [50] is a recently proposed variant of DMaps that defines similarity (or affinity) between very high-dimensional data points based on their underlying subspace structure and leverages DMaps to build a graph connecting subspaces on the Grassmann manifold [52,56–58]. Herein, the basic elements of GDMaps are briefly presented and drawn from [28,50].

2.1.1 Grassmann Manifold Principles

The *Grassmann manifold* or *Grassmannian*, denoted $\mathcal{G}(p,n)$ or $\mathcal{G}_{p,n}$, is the set of all p-dimensional subspaces embedded in $\mathbb{R}^n$. $\mathcal{G}(p,n)$ is a smooth manifold of dimension $p(n-p)$. A point on the Grassmannian, $\mathcal{X} \in \mathcal{G}(p,n)$, is represented (the Stiefel representation) by an orthonormal matrix $\mathbf{X} \in \mathbb{R}^{n\times p} : \mathbf{X}^T\mathbf{X} = \mathbf{I}_p$ where $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ is the identity matrix. That is, the point $\mathcal{X}$ is defined as the space spanned by the basis vectors $\mathbf{X}$, $\mathcal{X} = \text{span}(\mathbf{X})$.

For a group of points on the Grassmannian, the Riemannian center of mass, also known as the *Karcher mean* [29], is defined as the point $\mathcal{Y}$ that minimizes locally the cost function $\lambda : \mathcal{G}(p,n) \to \mathbb{R}_{\geq 0}$ given by

$$\lambda(\mathcal{Y}) = \int_{\mathcal{G}_{p,n}} d^2_{\mathcal{G}_{p,n}}(\mathcal{Y},\mathcal{X}) dP(\mathcal{X}), \tag{1}$$

where $dP(\mathcal{X}) = \rho(\mathcal{X}) d\mathcal{G}_{p,n}(\mathcal{X})$ is a probability measure over the infinitesimal volume element $d\mathcal{G}_{p,n}(\mathcal{X})$ with probability density $\rho(\mathcal{X})$ and $d_{\mathcal{G}_{p,n}}$ represents a distance measure on the Grassmannian. For a set of independent sample points $\{\mathcal{X}_i\}_{i=1}^N \in \mathcal{G}(p,n)$, the sample Karcher mean $\mathbf{m}$ is estimated as the local minimizer of

$$\lambda(\mathbf{m}) = \frac{1}{N}\sum_{i=1}^{N} d^2_{\mathcal{G}_{p,n}}(\mathcal{X}_i, \mathbf{m}). \tag{2}$$

Given the smoothness of the Grassmannian, one can define the *tangent space* at a given point $\mathcal{X} \in \mathcal{G}(p,n)$, denoted $\mathcal{T}_{\mathcal{X}}\mathcal{G}(p,n)$, as the derivative of a trajectory $\gamma(z)$ on the manifold. The tangent space is represented by the set of all tangent vectors in $\mathcal{X}$, such that

$$\mathcal{T}_{\mathcal{X}}\mathcal{G}(p,n) = \{\mathbf{\Gamma} \in \mathbb{R}^{n\times p} : \mathbf{\Gamma}^\top \mathbf{X} = 0\}. \tag{3}$$

The trajectory $\gamma(z)$ is defined as the shortest (geodesic) path between two points, $\mathcal{X}_0$ and $\mathcal{X}_1$ on $\mathcal{G}(p,n)$. If z is defined on the unit line, i.e., $z \in [0,1]$, the two points are denoted as $\gamma(0) = \mathcal{X}_0$ and $\gamma(1) = \mathcal{X}_1$.

In the neighborhood of a point $\mathcal{X}_0$, mapping between the Grassmannian and the tangent space can be performed by the *logarithmic* and *exponential* mappings. Consider two points $\mathcal{X}_0, \mathcal{X}_1$ on $\mathcal{G}(p,n)$ with $\gamma(0) = \mathcal{X}_0$, $\gamma(1) = \mathcal{X}_1$ represented by the matrices $\mathbf{X}_0$, $\mathbf{X}_1$, and $\dot\gamma(0) = \dot{\mathcal{X}}_0$ represented by the matrix $\mathbf{\Gamma}_0 \in \mathcal{T}_{\mathcal{X}_0}\mathcal{G}(p,n)$. One can map from a point $\mathcal{X}_1$ to the tangent space $\mathcal{T}_{\mathcal{X}_0}\mathcal{G}(p,n)$ through the logarithmic mapping,

$$\log_{\mathcal{X}_0}(\mathbf{X}_1) = \mathbf{\Gamma}_1 = \mathbf{U}\tan^{-1}(\mathbf{\Sigma})\mathbf{V}^\top. \tag{4}$$

where $\mathbf{\Gamma}_1 \in \mathcal{T}_{\mathcal{X}_0}\mathcal{G}(p,n)$ and $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ are obtained from the singular value decomposition (SVD) of the matrix $\mathbf{M} = (\mathbf{X}_1 - \mathbf{X}_0\mathbf{X}_0^\top\mathbf{X}_1)(\mathbf{X}_0^\top\mathbf{X}_1)^{-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$. Moreover, one can map from the point $\mathbf{\Gamma}_1 \in \mathcal{T}_{\mathcal{X}_0}\mathcal{G}(p,n)$ to the point $\mathbf{X}_1$ through the exponential mapping:

$$\exp_{\mathcal{X}_0}(\mathbf{\Gamma}_1) = \mathbf{X}_1 = \mathbf{X}_0\mathbf{V}\cos(\mathbf{\Sigma}) + \mathbf{U}\sin(\mathbf{\Sigma}), \tag{5}$$

where matrices $\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$ are defined by the SVD of $\mathbf{\Gamma}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Additional details can be found in [59].

Points on the Grassmannian are connected with smooth curves along which metrics of distances can be defined. Several such metrics exist [57]. Perhaps the most commonly used distance metric is the *geodesic* distance $d_{\mathcal{G}(p,n)}(\mathbf{X}_0,\mathbf{X}_1)$ between two points $\mathcal{X}_0, \mathcal{X}_1 \in \mathcal{G}(p,n)$, which corresponds to the distance over the geodesic $\gamma(t)$, $t \in [0,1]$, and is expressed as

$$d_{\mathcal{G}(p,n)}(\mathbf{X}_0,\mathbf{X}_1) = \|\mathbf{B}\|_2, \tag{6}$$

where $\mathbf{B} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the vector of principal angles obtained from the full SVD of $\mathbf{X}_0^\top\mathbf{X}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{B} = \cos^{-1}(\mathbf{\Sigma})$.

As we will see in the subsequent sections, a particularly useful way to analyze data on the Grassmannian is to embed the manifold into a Hilbert space using a valid kernel [60]. A Grassmannian kernel k is defined as the map

$$k : \mathcal{G}(p,n) \times \mathcal{G}(p,n) \to \mathbb{R}, \tag{7}$$

where k is positive semidefinite and invariant to the choice of basis. The notion of similarity is encoded by positive semidefinite kernels on a graph and is maximized when the distance is equal to zero. Several families of Grassmannian kernels exist in the literature [61], the most popular being the *Binet-Cauchy* and *projection* kernels. The Binet-Cauchy kernel is used to define the Plücker embedding which maps the Grassmann manifold $\mathcal{G}(p,n)$ to the projective space $\mathbb{P}(\bigwedge^p \mathbb{R}^n)$, where the exterior product $\bigwedge^p \mathbf{V}$ is the kth product of a vector space $\mathbf{V}$. The Binet-Cauchy kernel is defined as

$$k_{bc}(\mathbf{X}_0, \mathbf{X}_1) = \det(\mathbf{X}_0^\top \mathbf{X}_1)^2, \tag{8a}$$

$$k_{bc}(\mathbf{X}_0, \mathbf{X}_1) = \prod_{i=1}^{p} \cos^2(\beta_i), \tag{8b}$$

where Eq. (8b) expresses the relation between the kernel and the principal angles. Similarly, the projection kernel is defined using the projection embedding $\Pi : \mathcal{G}(p,n) \to \mathbb{R}^{n\times n}$ given by $\Pi(\mathbf{X}) = \mathbf{X}^T\mathbf{X}$. Finally, the projection kernel is defined as

$$k_p(\mathbf{X}_0, \mathbf{X}_1) = \|(\mathbf{X}_0^\top \mathbf{X}_1)\|_{\mathrm{F}}^2, \tag{9a}$$

$$k_p(\mathbf{X}_0, \mathbf{X}_1) = \sum_{i=1}^{p} \cos^2(\beta_i). \tag{9b}$$

Throughout this work, we use the projection kernel. The interested reader is referred to [50] for more information on how the Binet-Cauchy and projection kernels are constructed and applied.

2.1.2 Diffusion Maps on the Grassmannian

Consider a set of points (projected high-dimensional data) on the Grassmann manifold $\mathcal{G}(p,n)$ given by $\mathcal{G}_N = \{\mathcal{X}_1, ..., \mathcal{X}_N\}$ and a positive semidefinite Grassmannian kernel $k : \mathcal{G}(p,n)\times\mathcal{G}(p,n) \to \mathbb{R}$, also known as the diffusion kernel. If we consider a random walk over $\mathcal{G}_N$ having probability distribution f, $W_N = (\mathcal{G}_N, f, \mathbf{P})$, we can construct the transition probability matrix $\mathbf{P}$ as follows. First, we construct the degree matrix,

$$D_{ii} = \sum_{j=1}^{N} k(\mathcal{X}_i, \mathcal{X}_j), \tag{10}$$

where D_{ii} is a diagonal matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ and determine the stationary distribution of the random walk as

$$\pi_i = \frac{D_{ii}}{\sum_{k=1}^{N} D_{kk}}. \tag{11}$$

Next, the kernel is normalized as

$$\kappa_{ij} = \frac{k_{ij}}{\sqrt{D_{ii}D_{jj}}}, \tag{12}$$

and the transition probability matrix P_{ij} of the random walk over the Grassmannian is given by

$$P_{ij}^t = \frac{\kappa_{ij}}{\sum_{k=1}^{N} D_{ik}}. \tag{13}$$

Running the Markov chain forward in time is effectively equivalent to running a diffusion process on the manifold which allows us to reveal the geometric structure of the data on the Grassmannian. From the eigendecomposition

of $\mathbf{P}^t$ we find the truncated diffusion map basis consisting of the first q eigenvectors $\{\xi_k\}_{k=1}^q$, with $\xi_k \in \mathbb{R}^N$ and corresponding eigenvalues $\{\lambda_k\}_{k=1}^q$. Therefore, the diffusion coordinates are defined as

$$\mathbf{\Theta}_j = (\theta_{j0}, ..., \theta_{jq}) = (\lambda_0 \xi_{j0}, ..., \lambda_q \xi_{jq}), \tag{14}$$

where ξ_{jk} corresponds to the position j of ξ_k. Due to the spectral decay of the eigenvalues of the sparse Markov matrix, usually a small q is sufficient to capture the essential geometric structure of the dataset.

We note here that there are two essential features that distinguish the GDMaps from the conventional DMaps:

1. **Data points lie on $\mathcal{G}(p, n)$:** The data on which DMaps is performed are, in fact, subspaces that compactly span the space in which the original data lie.

2. **A Grassmannian kernel is employed:** The Grassmannian kernel is an effective means of assessing the similarity between subspaces.

The motivation to use a subspace representation is primarily related to the difficulty in assessing similarity between very high-dimensional objects and is further elaborated in [50].

2.2 Geometric Harmonics (GH)

Introduced by Coifman and Lafon [53] and based on the Nyström method, GH is a method for extending an empirical function defined on a set X to a set $\bar{X}$, where $X \subset \bar{X}$. This out-of-sample extension scheme aims to deal with the limitations of similar techniques (e.g., kriging) related to the choice of a scale of extension. If we assume that a real-valued function $f : X \to \mathbb{R}$ is defined on X, GH provides a way to find an extension of f, say a new function $F : \bar{X} \to \mathbb{R}$.

To begin, consider a symmetric, positive semidefinite, and bounded kernel $k : \bar{X} \times \bar{X} \to \mathbb{R}$, which defines a unique reproducing kernel Hilbert space $\mathcal{H}$ of functions defined on $\bar{X}$, for which k is the reproducing kernel. A typical choice is the Gaussian kernel, expressed by

$$K_{ij} = k(x_i, x_j) = \exp\left(- \frac{\|x_i - x_j\|_2^2}{\epsilon^2} \right), \tag{15}$$

where ϵ is a tunable length scale, $i, j = 1, ..., \mathcal{N}$, and $\| \cdot \|_2$ is the Euclidean norm.

Given the above, it is possible to represent the function f in terms of the eigenfunctions of k and then extend it for out-of-sample predictions [62]. Given $\mathcal{N}$ realizations of the function $f(x)$, denoted $Y = \{y_i\} = \{f(x_i)\}, i = 1, \ldots, \mathcal{N}$, evaluated at a set of sample points $X = \{x_i\}, i = 1, \ldots, \mathcal{N}$, we evaluate the kernel matrix $\mathbf{K}$ with elements $K_{ij} = k(x_i, x_j)$ and perform an eigen decomposition to obtain l eigenvalues, $\mathbf{\Lambda} = \text{diag}(\lambda_i), i = 1, \ldots, l$ and eigenvectors $\mathbf{\Psi} = [\psi_i], i = 1, \ldots, l$ with $l = \max_i \mid \lambda_i \geq \delta\lambda_0$. We then project the function f onto the space spanned by $\psi_i, i = 1, \ldots, l$:

$$f \mapsto \mathbf{P}_\delta f = \sum_{j=1}^{l} \langle f, \psi_j \rangle_X \psi_j. \tag{16}$$

Practically, this is achieved by projecting the points Y as $Y_\delta = \mathbf{\Psi}^\intercal Y$.

We then apply the Nyström extension to extend the discrete eigenvectors on X to eigenfunctions on $\bar{X}$ corresponding to $\mathcal{N}_*$ out-of-sample points $\bar{x}_i, i = 1, \ldots, \mathcal{N}_*$. To do so, we build the extended kernel matrix as $\bar{\mathbf{K}}$ having elements $\bar{K}_{ij} = k(\bar{x}_i, x_j)$. The extended eigenvectors, or geometric harmonics, are then expressed in the matrix $\mathbf{\Phi}$ whose pth component evaluated at extension point $\bar{x}_i$ is given by

$$\phi_i^{(p)} = \phi^{(p)}(\bar{x}_i) = \frac{1}{\lambda_p} \sum_{j=1}^{\mathcal{N}} \bar{K}_{ij} \psi_j^{(p)}. \tag{17}$$

Again, for practical implementation, the GH matrix is constructed simply as $\mathbf{\Phi} = \bar{\mathbf{K}}\mathbf{\Psi}\mathbf{\Lambda}^{-1}$.

Finally, we extend the function $\mathbf{P}_\delta f$ on X in basis ψ_j to basis ϕ_j on $\bar{X}$ as

$$\mathbf{E}f(\bar{x}) = \sum_{j=1}^{l} \langle f, \psi_j \rangle_X \phi_j(\bar{x}), \tag{18}$$

which can be performed for out-of-sample extension to a set of predicted points $\bar{Y}$ as

$$\bar{Y} = \mathbf{\Phi} Y_\delta = \mathbf{\Phi} = \bar{\mathbf{K}}\mathbf{\Psi}\mathbf{\Lambda}^{-1}\mathbf{\Psi}^\intercal Y. \tag{19}$$

For more details and examples on GH, the reader is referred to [53].

2.3 Polynomial Chaos Expansion (PCE)

We assume a model denoted as $\mathcal{M}(\mathbf{X})$, $\mathbf{X}$ being a k-variate random variable defined on the probability space (Ω, Σ, P) and characterized by the joint probability density function (PDF) $\varrho_{\mathbf{X}} : Z \to \mathbb{R}_{\geq 0}$, where $Z \subseteq \mathbb{R}^k$ is the image space, Ω the sample space, Σ the set of events, and P the probability measure. Throughout this work it is assumed that $\mathbf{X}$ consists of independent random variables; however, we note that the PCE method is applicable for the case of dependent random variables as well; see e.g., [63–65]. Then, assuming that model $\mathcal{M}$ satisfies the conditions of the Doob-Dynkin lemma [66], its output $\mathcal{M}(\mathbf{X})$ is a random variable dependent on $\mathbf{X}$. In the following, we consider for simplicity a single model output, such that $Y(\omega) = \mathcal{M}(\mathbf{X}(\omega)) \in \mathbb{R}$, $\omega \in \Omega$. Nevertheless, the extension to multivariate outputs is straightforward, as the PCE approximation described next can be applied elementwise. Note that in the following we use the same notation for a random variable $\mathbf{X}$ and a realization $\mathbf{X}(\omega)$; however, the distinction between the two should be clear from the context.

Under the assumption of a single model output, the PCE is a spectral approximation of the form

$$\mathcal{M}(\mathbf{X}) \approx \widetilde{\mathcal{M}}(\mathbf{X}) = \sum_{s=1}^{S} c_s \Xi_s(\mathbf{X}), \tag{20}$$

where c_s are scalar coefficients and Ξ_s are multivariate polynomials that are orthonormal with respect to the joint PDF $\varrho_{\mathbf{X}}$, such that

$$\mathbb{E}[\Xi_s \Xi_t] = \int_Z \Xi_s(\mathbf{X}) \Xi_t(\mathbf{X}) \varrho_{\mathbf{X}}(\mathbf{X}) \mathrm{d}\mathbf{X} = \delta_{st}, \tag{21}$$

where δ_{st} denotes the Kronecker delta. Depending on the PDF $\varrho_{\mathbf{X}}$, the orthonormal polynomials can be chosen according to the Wiener-Askey scheme [2] or be numerically constructed [67,68]. Since $\mathbf{X}$ is assumed to consist of independent random variables $X_1, \ldots, X_k$, the joint PDF is given as

$$\varrho_{\mathbf{X}}(\mathbf{X}) = \prod_{i=1}^{k} \varrho_{X_i}(X_i), \tag{22}$$

where ϱ_{X_i} is the marginal PDF of random variable X_i. Accordingly, the multivariate orthogonal polynomials are constructed as

$$\Xi_s(\mathbf{X}) \equiv \Xi_{\mathbf{s}}(\mathbf{X}) = \prod_{i=1}^{k} \xi_i^{s_i}(X_i), \tag{23}$$

where $\xi_i^{s_i}$ are univariate polynomials of degree $s_i \in \mathbb{Z}_{\geq 0}$ and orthonormal with respect to the univariate PDF ϱ_{X_i}, such that

$$\mathbb{E}\left[\xi_i^{s_i} \xi_i^{t_i}\right] = \int_{Z_i} \xi_i^{s_i}(X_i) \xi_i^{t_i}(X_i) \varrho_{X_i}(X_i) \, \mathrm{d}X_i = \delta_{s_i t_i}. \tag{24}$$

The multi-index $\mathbf{s} = (s_1, \ldots, s_k)$ is equivalent to the multivariate polynomial degree and uniquely associated to the single index s employed in Eq. (20), which can now be written in the equivalent form,

$$\mathcal{M}(\mathbf{X}) \approx \widetilde{\mathcal{M}}(\mathbf{X}) = \sum_{\mathbf{s}\in\Lambda} c_{\mathbf{s}} \Xi_{\mathbf{s}}(\mathbf{X}), \tag{25}$$

where Λ is a multi-index set with cardinality $\#\Lambda = S$. The choice of the multi-index set Λ plays a central role in the construction of the PCE, as it defines which polynomials and corresponding coefficients form the PCE. The most common choice, as well as the one employed in this work, is that of a total-degree multi-index set, such that Λ includes all multi-indices that satisfy $\|\mathbf{s}\|_1 \leq s_{\max}$, $s_{\max} \in \mathbb{Z}_{\geq 0}$. In that case, the size of the PCE basis is $S = (s_{\max} + k)!/s_{\max}!k!$, i.e., it scales polynomially with the input dimension k and the maximum degree $s_{\max}$. For the case of high-dimensional input random variables $\mathbf{X}$, several sparse PCE algorithms have been proposed in the literature for the construction of Λ such that the impact of the curse of dimensionality is mitigated [10,12–14,69,70].

Once the multi-index set Λ is fixed, the only thing remaining to complete the PCE is to compute the coefficients. Several approaches are suggested in the literature for computing the PCE coefficients, e.g., pseudo-spectral projection [71–74], interpolation [75,76], and, most commonly, regression [4,10,13,14,69,70,77,78]. The latter option is employed in this work also, such that the PCE coefficients are obtained by solving the penalized least squares problem [79],

$$\underset{\mathbf{c}\in\mathbb{R}^{\#\Lambda}}{\arg\min} \left\{ \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} \left(\mathcal{M}(\mathbf{X}_i) - \sum_{\mathbf{s}\in\Lambda} c_{\mathbf{s}} \Xi_{\mathbf{s}}(\mathbf{X}_i) \right)^2 + \lambda J(\mathbf{c}) \right\}, \tag{26}$$

where $\lambda \in \mathbb{R}$ is a penalty factor, $J(\mathbf{c})$ a penalty function acting on the vector of PCE coefficients $\mathbf{c} \in \mathbb{R}^{\#\Lambda}$, and $\mathscr{X} = \{\mathbf{X}_i\}_{i=1}^{\mathcal{N}}$ an experimental design (ED) of random variable realizations with corresponding model outputs $\mathscr{Y} = \{\mathbf{Y}_i\}_{i=1}^{\mathcal{N}}$. Common choices for the penalty function $J(\mathbf{c})$ are the ℓ_1 and ℓ_2 norms, in which case problem (26) is referred to as LASSO (least absolute shrinkage and selection operator) and ridge regression, respectively. Removing the penalty term results in an ordinary least squares (OLS) regression problem.

3. HIGH-DIMENSIONAL SURROGATES USING PCE ON THE GRASSMANNIAN DIFFUSION MANIFOLD

In the next sections, the essential ingredients of the proposed method for constructing surrogate models on lower-dimensional manifolds are analytically presented. The proposed approach is composed of two paths: (1) an encoder path where high-dimensional model data are embedded onto a low-dimensional Grassmannian diffusion manifold and a PCE surrogate is constructed to map from the input space to the low-dimensional latent space and (2) a decoder path in which predicted low-dimensional solutions in the latent space are expanded to reconstruct corresponding full, high-dimensional solutions. The approach is illustrated graphically in Fig. 1.

3.1 Encoder Path

The encoder path to produce an inexpensive mapping from the input space to a low-dimensional latent space representation of the high-dimensional response is detailed herein and the corresponding algorithm is provided in Algorithm 1.

3.1.1 Training Realizations

Consider an ED $\mathscr{X} = \{\mathbf{X}_1, ..., \mathbf{X}_{\mathcal{N}}\}$ with $\mathcal{N}$ i.i.d. random samples $\mathbf{X}_i \in \mathbb{R}^k$ drawn from the joint PDF $\varrho_{\mathbf{X}}$. A model $\mathcal{M} : \mathbf{X}_i \in \mathbb{R}^k \rightarrow \mathbf{Y}_i \in \mathbb{R}^{n\times m}$ (analytical or computational) is then used to generate the corresponding model evaluations $\mathscr{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_{\mathcal{N}}\}$. We assume that the dimensionality of the quantity of interest (QoI) $\mathbf{Y}$ is high, e.g., on the order of $\mathcal{O}(10^{4-6})$ corresponding, for example, to the number of degrees of freedom in the system, the number of time instants over which the solution is obtained, or both. We further assume that a train-test splitting procedure has already been performed for the evaluation of the surrogate model on previously unseen data. Thus, $\mathcal{N}$ represents

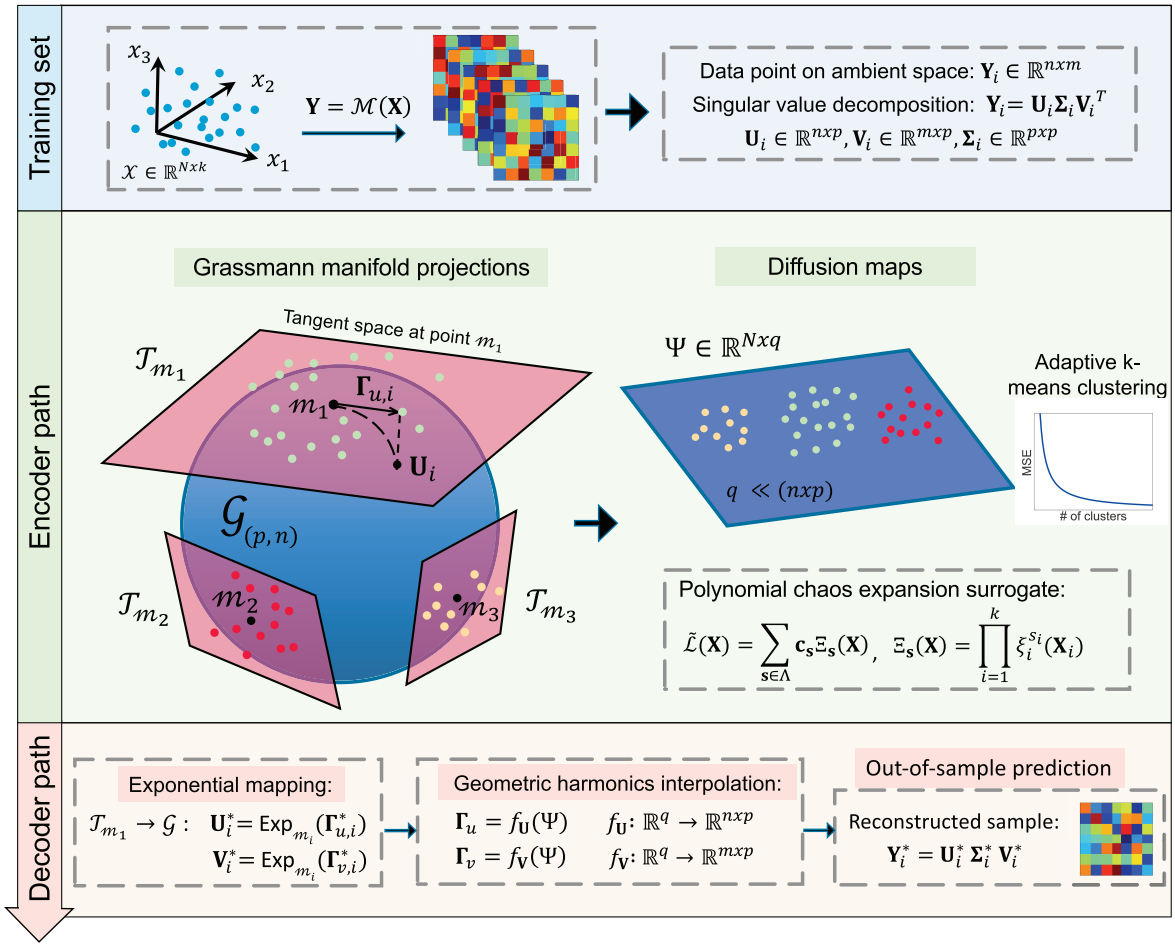

FIG. 1: A schematic illustrating the proposed encoder-decoder framework for constructing PCE surrogates on Grassmannian diffusion manifolds. The Grassmannian $\mathcal{G}(p,n)$ on which the set of $\{\mathbf{U}_i\}_{i=1}^{\mathcal{N}}$ matrices lives, is depicted. A second Grassmannian $\mathcal{G}(p,m)$ exists, on which the $\{\mathbf{V}_i\}_{i=1}^{\mathcal{N}}$ matrices live. The diffusion manifold, represented by the diffusion coordinates $\{\mathbf{\Theta}_i\}_{i=1}^{\mathcal{N}}$, is also depicted. Local GH models are used to identify inverse mappings between the diffusion coordinates and points $\{\mathbf{\Gamma}_{u,i}, \mathbf{\Gamma}_{v,i}\}_{i=1}^{\mathcal{N}}$ that live on the various tangent spaces $\mathcal{T}_{m_i}$. Finally, exponential mappings and reverse SVD are used for sample reconstruction.

Algorithm 1: Polynomial chaos expansion on diffusion manifolds (encoder path)

input : ED $\mathscr{X} = \{\mathbf{X}_1, ..., \mathbf{X}_\mathcal{N}\}$ where $\mathbf{X}_i \in \mathbb{R}^k$ and $\mathscr{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_\mathcal{N}\}$ where $\mathbf{Y}_i \in \mathbb{R}^{n\times m}$ via model $\mathcal{M}(\mathbf{X})$

output: Diffusion coordinates $\Theta = \{\mathbf{\Theta}_1, ..., \mathbf{\Theta}_\mathcal{N}\}$ where $\mathbf{\Theta}_i \in \mathbb{R}^q$ and PCE surrogate $\widetilde{\mathcal{L}}$

for $i \leftarrow 1$ **to** $\mathcal{N}$ **do**
  Perform SVD $\mathbf{Y}_i = \mathbf{U}_i\mathbf{\Sigma}_i{\mathbf{V}_i}^\top$ where $\mathbf{U}_i \in \mathcal{G}(p,n)$ and $\mathbf{V}_i \in \mathcal{G}(p,m)$
end
Construct a Grassmannian diffusion kernel $k(\mathbf{U}, \mathbf{V})$, e.g., the Binet-Cauchy via Eq. (8)
Obtain the diffusion coordinates $\{\mathbf{\Theta}_i \in \mathbb{R}^q\}_{i=1}^{\mathcal{N}}$ [with Eqs. (13) and (14)] where $q \ll n \times m$
Construct PCE approximation $\widetilde{\mathcal{L}}(\mathbf{X}) = \sum_{\mathbf{s}\in\Lambda} \mathbf{c_s}\Xi_\mathbf{s}(\mathbf{X})$ where $\mathbf{c_s} \in \mathbb{R}^q$ can be computed via Eq. (26)

the number of training samples which, for most real-world engineering applications, is rather small, e.g., on the order of $\mathcal{O}(10^{1-3})$.

3.1.2 Grassmannian Diffusion Manifold Projection

As a first step, we perform linear dimension reduction by projecting the high-dimensional data points onto the Grassmann manifold. Prior to the projection, each data point is reshaped to as close to a square matrix as possible, which allow us to perform significant dimension reduction while preserving the most essential information. For the set of $\mathcal{N}$ data $\mathscr{Y} = \{\mathbf{Y}_1, .., \mathbf{Y}_\mathcal{N}\}$, $\mathbf{Y}_i \in \mathbb{R}^{n\times m}$, we project each data point $\mathbf{Y}_i$ onto the Grassmannian by performing a thin singular value decomposition (SVD) as

$$\mathbf{Y}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^\top, \tag{27}$$

where the columns of the matrices (subspaces) $\mathbf{U}_i \in \mathbb{R}^{n\times p}$ and $\mathbf{V}_i \in \mathbb{R}^{m\times p}$ contain orthonormal singular vectors such that $\mathbf{U}_i^\top \mathbf{U}_i = \mathbf{I}_p$ and $\mathbf{V}_i^\top \mathbf{V}_i = \mathbf{I}_p$, and $\mathbf{\Sigma}_i \in \mathbb{R}^{p\times p}$ is a diagonal matrix whose nonzero elements are the singular values ordered by magnitude. Therefore, $\mathbf{U}_i, \mathbf{V}_i$ live on the Grassmannians $\mathcal{G}(p,n) = \{\text{span}(\mathbf{U}) : \mathbf{U} \in \mathbb{R}^{n\times p}\}$ and $\mathcal{G}(p,m) = \{\text{span}(\mathbf{V}) : \mathbf{V} \in \mathbb{R}^{m\times p}\}$, respectively. The value of dimension p is either specified *a priori*, or computed automatically by assigning a tolerance for the SVD.

Next, for every pair $[\mathbf{U}_i, \mathbf{U}_j]$ and $[\mathbf{V}_i, \mathbf{V}_j]$ we compute the entries of k_{ij} of the kernel matrices $k_{ij}(\mathbf{U})$ and $k_{ij}(\mathbf{V})$. We choose to construct either the Binet-Cauchy kernel in Eq. (8) or the projection kernel in Eq. (9); thus the mappings are defined as $k_{ij}(\mathbf{U}) : \mathcal{G}(p,n) \times \mathcal{G}(p,n) \to \mathbb{R}$ and $k_{ij}(\mathbf{V}) : \mathcal{G}(p,m) \times \mathcal{G}(p,m) \to \mathbb{R}$, respectively. We then compute the composed kernel matrix $K(\mathbf{U}, \mathbf{V})$ either by taking the sum or product of the corresponding kernels, i.e.,

$$K(\mathbf{U}, \mathbf{V}) = K(\mathbf{U}) + K(\mathbf{V}), \tag{28a}$$

$$K(\mathbf{U}, \mathbf{V}) = K(\mathbf{U}) \circ K(\mathbf{V}), \tag{28b}$$

where $\circ$ denotes the Hadamard product. The composed kernel $K(\mathbf{U}, \mathbf{V})$, having components k_{ij}, is then used to construct the diagonal matrix $\mathbf{D} \in \mathbb{R}^{N\times N}$ in Eq. (10) and then the normalized matrix κ with components κ_{ij} in Eq. (12). Next we construct the transition probability matrix $\mathbf{P}^t$ of the Markov chain over the data and we perform an eigendecomposition of $\mathbf{P}^t$ to determine the truncated diffusion map basis of q eigenvectors $\{\xi_k\}_{k=1}^q$, with $\xi_k \in \mathbb{R}^N$ and corresponding eigenvalues $\{\lambda_k\}_{k=1}^q$. The diffusion coordinates are therefore given by $\Theta = \{\mathbf{\Theta}_1, ..., \mathbf{\Theta}_\mathcal{N}\}$ where $\mathbf{\Theta}_i \in \mathbb{R}^q$. Herein, we will refer to the mapping of the input parameters $\mathbf{X}$ to the diffusion coordinates Θ as $\mathcal{L} : \mathbf{X}_i \in \mathbb{R}^k \to \mathbf{\Theta}_i \in \mathbb{R}^q$.

The dimension q of the diffusion coordinates $\mathbf{\Theta}_i \in \mathbb{R}^q$ (embedding) is much smaller than the dimension of the data on the ambient space $\mathbf{Y}_i \in \mathbb{R}^{n\times m}$ (i.e., $q \ll n \times m$) and therefore GDMaps allows us to achieve a significant dimension reduction.

3.1.3 Surrogate Modeling via PCE

Given a training dataset of input random variable realizations $\mathscr{X} = \{\mathbf{X}_1, ..., \mathbf{X}_\mathcal{N}\}$, $\mathbf{X}_i \in \mathbb{R}^k$, and corresponding solutions projected on the latent space $\Theta = \{\mathbf{\Theta}_1, ..., \mathbf{\Theta}_\mathcal{N}\}$, $\mathbf{\Theta}_i \in \mathbb{R}^q$, we construct a PCE as explained in Section 2.3 to approximate the true encoder $\mathcal{L} : \mathbf{X} \to \mathbf{\Theta}$ as

$$\widetilde{\mathcal{L}}(\mathbf{X}) = \sum_{\mathbf{s}\in\Lambda} \mathbf{c}_\mathbf{s} \Xi_\mathbf{s}(\mathbf{X}), \tag{29}$$

where Λ is a total-degree multi-index set, $\Xi_\mathbf{s}$ are the multivariate orthonormal polynomials, and the PCE coefficients are now vector-valued with dimension equal to the one of the diffusion coordinates, i.e., $\mathbf{c}_\mathbf{s} \in \mathbb{R}^q$.

To assess the predictive ability of the PCE surrogate, we employ an error metric known as the *generalization error* [9], which is defined as

$$\epsilon_{\text{gen}} = \mathbb{E}_\mathbf{X}\Big[\big(\mathcal{L}(\mathbf{X}) - \widetilde{\mathcal{L}}(\mathbf{X})\big)^2\Big]. \tag{30}$$

We approximate ϵ_{gen} with the *validation error*, which is computed on a validation dataset of $\mathcal{N}_*$ test realizations. The validation error is computed as

$$\epsilon_{\text{val}} = \frac{\sum_{i=1}^{\mathcal{N}_*} \left(\boldsymbol{\Theta}_i^* - \widetilde{\mathcal{L}}(\mathbf{X}_i^*)\right)^2}{\sum_{i=1}^{\mathcal{N}_*} \left(\boldsymbol{\Theta}_i^* - \bar{\boldsymbol{\Theta}}^*\right)^2}, \tag{31}$$

where $\{\mathbf{X}_i^* \in \mathbb{R}^k\}_{i=1}^{\mathcal{N}_*}$, $\{\boldsymbol{\Theta}_i^* \in \mathbb{R}^q\}_{i=1}^{\mathcal{N}_*}$, and $\bar{\boldsymbol{\Theta}}^* = (1/\mathcal{N}_*)\sum_{i=1}^{\mathcal{N}_*} \boldsymbol{\Theta}_i^*$ is the mean response. Accordingly, the total degree $s_{\max}$ is chosen so that the validation error is minimized. In cases where a validation dataset cannot be generated due to computational constraints, alternative measures such as the *k-fold cross validation* can be considered [9,12, 13]. However, such techniques introduce different computational costs, as they require the construction of multiple surrogates for different partitionings of the training dataset; therefore, a prior evaluation of the trade-off and respective costs is required.

3.2 Decoder Path

Given predictions from the PCE in the low-dimensional latent space (Grassmannian diffusion manifold), we reconstruct the approximate high-dimensional solution using the decoder path described below and detailed in Algorithm 2.

Algorithm 2: Polynomial chaos expansion on diffusion manifolds method (decoder path)

input : Testing samples $\mathscr{X}^* = \{\mathbf{X}_1, .., \mathbf{X}_{\mathcal{N}_*}\}$
output: Predicted solutions $\mathscr{Y}^* = \{\mathbf{Y}_1, .., \mathbf{Y}_{\mathcal{N}}\}$ for testing samples $\mathscr{X}^*$

$\ell = 2$
while $\mathcal{N}_{min}^h > 5$ *and* $\epsilon_t > 10^{-2}$ **do**
  Perform k-means clustering to identify clusters $\{C_h\}_{h=1}^{\ell}$ with $\mathcal{N}_h$ points, where $h = 1, .., \ell$
  Compute the Karcher means $\mathbf{m}_{u,h}, \mathbf{m}_{v,h}$ from Eq. (2), of points $\{\mathbf{U}_i\}_{i=1}^{\mathcal{N}_h} \in C_h$ and $\{\mathbf{V}_i\}_{i=1}^{\mathcal{N}_h} \in C_h$
  Having as origin $\mathbf{m}_{u,h}, \mathbf{m}_{v,h}$, project points onto $\mathcal{T}_{\mathbf{m}_{u,h}}$, $\mathcal{T}_{\mathbf{m}_{v,h}}$ via logarithmic mapping with Eq. (4)
  Project back to the Grassmannian via exponential mapping with Eq. (5)
  Compute $\text{MSE}_{\mathbf{U}}$, $\text{MSE}_{\mathbf{V}}$ with Eq. (33)
  Compute total error ϵ_t
  $\ell \leftarrow \ell + 1$
end
Construct 2ℓ GH models $\{f_{\mathbf{U}}^{C_h}\}_{h=1}^{\ell} : \mathbb{R}^q \rightarrow \mathbb{R}^{n\times p}$ and $\{f_{\mathbf{V}}^{C_h}\}_{h=1}^{\ell} : \mathbb{R}^q \rightarrow \mathbb{R}^{m\times p}$ to map diff. coordinates $\{\boldsymbol{\Theta}_i\}_{i=1}^{\mathcal{N}}$ and points on the tangent space $\{\boldsymbol{\Gamma}_{u,i}, \boldsymbol{\Gamma}_{v,i}\}_{i=1}^{\mathcal{N}}$ of the Grassmannian, based on Section 3.2.1
Construct PCE surrogate to map $\{\boldsymbol{\Theta}_i\}_{i=1}^{\mathcal{N}}$ to singular values $\{\boldsymbol{\Sigma}_i \in \mathbb{R}^{p\times p}\}_{i=1}^{\mathcal{N}}$
Out-of-sample prediction
Sample new points $\mathscr{X}^* = \{\mathbf{X}_1, .., \mathbf{X}_{\mathcal{N}_*}\}$
for $i \leftarrow 1$ **to** $\mathcal{N}_*$ **do**
  Compute the diffusion coordinates $\boldsymbol{\Theta}_i^*$ with PCE model $\widetilde{\mathcal{L}}$
  Identify the cluster C_h in which point i belongs to (using a Euclidean metric)
  Compute $\boldsymbol{\Gamma}_{u,C_h,i}^* = f_{\mathbf{U}}^{C_h}(\boldsymbol{\Theta}_{C_h,i}^*)$ and $\boldsymbol{\Gamma}_{v,C_h,i}^* = f_{\mathbf{V}}^{C_h}(\boldsymbol{\Theta}_{C_h,i}^*)$
  Perform exponential mapping to obtain $\mathbf{U}_i^* \in \mathbb{R}^{n\times p}$ and $\mathbf{V}_i^* \in \mathbb{R}^{m\times p}$
  Use PCE of singular values to obtain $\boldsymbol{\Sigma}_i^* \in \mathbb{R}^{p\times p}$
  Obtain predicted solution as $\mathbf{Y}_i^* = \mathbf{U}_i^*\boldsymbol{\Sigma}_i^*\mathbf{V}_i^*$
end
Obtain the set of predicted solutions $\mathscr{Y}^* = \{\mathbf{Y}_1, .., \mathbf{Y}_{\mathcal{N}_*}\}$

3.2.1 Adaptive Clustering of Solutions on the Manifold

An adaptive technique is proposed to cluster the diffusion coordinates $\{\boldsymbol{\Theta}_i\}_{i=1}^{\mathcal{N}}$ to iteratively identify an optimal number of clusters ℓ, such that the distance between points belonging to one cluster is minimized both on the diffusion manifold and on the Grassmannian. We note that since the diffusion maps basis is constructed with the use of a Grassmannian kernel (affinity matrix), the notion of similarity on the Grassmannian is preserved such that points that are close on the Grassmannian are similarly close on the diffusion manifold.

To partition points at each iteration we use the k-means clustering algorithm. We begin with the smallest possible number of clusters, i.e., $\ell = 2$. At each iteration and for each cluster C_h where $h = 1, .., \ell$, we compute the Karcher means $\mathbf{m}_{u,h}, \mathbf{m}_{v,h}$ from Eq. (2), of points $\{\mathbf{U}_i\}_{i=1}^{\mathcal{N}_h} \in C_h$ and $\{\mathbf{V}_i\}_{i=1}^{\mathcal{N}_h} \in C_h$, respectively, where $\mathcal{N}_h$ represents the total number of points for a given cluster. For the computation of the Karcher means and the minimization of the loss function we use stochastic gradient descent [80].

Having as origin the Karcher means $\mathbf{m}_{u,h}, \mathbf{m}_{v,h}$, we project the points of each cluster to the corresponding tangent spaces $\mathcal{T}_{\mathbf{m}_{u,h}}$, $\mathcal{T}_{\mathbf{m}_{v,h}}$ via the logarithmic mapping in Eq. (4). Next we project the same points back onto the Grassmannian via the exponential mapping in Eq. (5). This procedure can be expressed as

$$\{\mathbf{U}_i \in \mathcal{G}_{p,n}\}_{i=1}^{\mathcal{N}_h} \rightarrow \{\boldsymbol{\Gamma}_{u,i} \in \mathcal{T}_{\mathbf{m}_{u,h}}(\mathcal{G}_{p,n})\}_{i=1}^{\mathcal{N}_h} \rightarrow \{\widetilde{\mathbf{U}}_i \in \mathcal{G}_{p,n}\}_{i=1}^{\mathcal{N}_h}, \tag{32a}$$

$$\{\mathbf{V}_i \in \mathcal{G}_{p,n}\}_{i=1}^{\mathcal{N}_h} \rightarrow \{\boldsymbol{\Gamma}_{v,i} \in \mathcal{T}_{\mathbf{m}_{v,h}}(\mathcal{G}_{p,m})\}_{i=1}^{\mathcal{N}_h} \rightarrow \{\widetilde{\mathbf{V}}_i \in \mathcal{G}_{p,m}\}_{i=1}^{\mathcal{N}_h}, \tag{32b}$$

where the subscript h corresponds to a given cluster. Since the logarithmic and exponential mappings are only accurate in the local neighborhood of their origin (Karcher mean), the projection introduces error for points that deviate significantly from the Karcher mean. The pointwise error of the mapping of points $\{\mathbf{U}_i\}_{i=1}^{\mathcal{N}_h}$ and $\{\mathbf{V}_i\}_{i=1}^{\mathcal{N}_h}$ to the tangent spaces and back is computed via the mean-squared error (MSE) as

$$\text{MSE}_{\mathbf{U}} = \frac{1}{\mathcal{N}_h}\sum_{i=1}^{\mathcal{N}_h}(\mathbf{U}_i - \widetilde{\mathbf{U}}_i)^2, \tag{33a}$$

$$\text{MSE}_{\mathbf{V}} = \frac{1}{\mathcal{N}_h}\sum_{i=1}^{\mathcal{N}_h}(\mathbf{V}_i - \widetilde{\mathbf{V}}_i)^2, \tag{33b}$$

respectively. Finally, the total error ϵ_t at each iteration is computed as the average of errors corresponding to each cluster. A cluster of points that are not "close" on the Grassmannian will result in a significant error.

We repeat this process until the error is minimized or until a specified minimum number of points has been detected in a cluster (usually $\mathcal{N}_{\text{min}}^h \simeq 5 - 10$). Alternatively, the process can stop when the total error is below a predefined threshold, e.g., $\epsilon_t < 10^{-2}$. Once the optimal number of clusters ℓ has been identified and the data on the manifold $\{\boldsymbol{\Theta}_i\}_{i=1}^{\mathcal{N}}$ have been appropriately partitioned in $\{C_h\}_{h=0}^{\ell}$ clusters, out-of-sample predictions can be performed as described in the following section.

3.2.2 Out-of-Sample Extension for High-Dimensional Solution Prediction

Consider $\mathcal{N}_*$ additional realizations of points on the diffusion manifold $\{\boldsymbol{\Theta}_i^* \in \mathbb{R}^q\}_{i=1}^{\mathcal{N}_*}$ have been generated using the PCE surrogate. We propose the following inverse map framework to return samples to the physically interpretable space and compute the reconstructed samples $\{\mathbf{Y}_i^* \in \mathbb{R}^{n\times m}\}_{i=1}^{\mathcal{N}_*}$.

First, we identify a mapping between training data on the diffusion manifold $\{\boldsymbol{\Theta}_i\}_{i=1}^{\mathcal{N}}$ and corresponding points on the tangent spaces of the Grassmannian $\{\boldsymbol{\Gamma}_{u,i}, \boldsymbol{\Gamma}_{v,i}\}_{i=1}^{\mathcal{N}}$ by constructing 2ℓ local geometric harmonics (GH) models as described in Section 2.2, two for each cluster C_h, $h = 1, \ldots, \ell$, as follows:

$$\{f_{\mathbf{U}}^{C_h}\}_{h=1}^{\ell} : \mathbb{R}^q \rightarrow \mathbb{R}^{n\times p}, \tag{34a}$$

$$\{f_{\mathbf{V}}^{C_h}\}_{h=1}^{\ell} : \mathbb{R}^q \rightarrow \mathbb{R}^{m\times p}, \tag{34b}$$

where $f_{\mathbf{U}}^{C_h}$ and $f_{\mathbf{V}}^{C_h}$ correspond to the mapping of diffusion coordinates Θ^*, to matrices $\mathbf{\Gamma}_u^*, \mathbf{\Gamma}_v^*$, respectively, for cluster C_h. More specifically, using the GH models, we compute the tangent space matrices as

$$\mathbf{\Gamma}_{u,C_h,i}^* = f_{\mathbf{U}}^{C_h}(\mathbf{\Theta}_{C_h,i}^*), \quad i = 1, \ldots, \mathcal{N}_*, \tag{35a}$$

$$\mathbf{\Gamma}_{v,C_h,i}^* = f_{\mathbf{V}}^{C_h}(\mathbf{\Theta}_{C_h,i}^*), \quad i = 1, \ldots, \mathcal{N}_*, \tag{35b}$$

where $\mathbf{\Gamma}_{u,C_h,i}^*, \mathbf{\Gamma}_{v,C_h,i}^*$ reside on the tangent spaces $\mathcal{T}_{C_h}\mathcal{G}(p,n)$ and $\mathcal{T}_{C_h}\mathcal{G}(p,m)$, respectively, for cluster C_h.

Once the points on the tangent spaces $\{\mathbf{\Gamma}_{u,C_h,i}^* \in \mathbb{R}^{n\times p}\}_{i=1}^{\mathcal{N}_*}$ and $\{\mathbf{\Gamma}_{v,C_h,i}^* \in \mathbb{R}^{m\times p}\}_{i=1}^{\mathcal{N}_*}$ have been computed, the exponential mapping in Eq. (5) is used to project onto the points $\{\mathbf{U}_i^* \in \mathbb{R}^{n\times p}\}_{i=1}^{\mathcal{N}_*}$ and $\{\mathbf{V}_i^* \in \mathbb{R}^{m\times p}\}_{i=1}^{\mathcal{N}_*}$ on the Grassmannians $\mathcal{G}_{p,n}$ and $\mathcal{G}_{p,m}$ (where here we drop the C_h subscripts for simplicity). Finally, we construct a global PCE surrogate to map between diffusion coordinates $\{\mathbf{\Theta}_i\}_{i=1}^{\mathcal{N}}$ and the diagonal singular value matrices $\{\mathbf{\Sigma}_i \in \mathbb{R}^{p\times p}\}_{i=1}^{\mathcal{N}}$, which can be used to estimate new realizations of the singular values $\{\mathbf{\Sigma}_i^* \in \mathbb{R}^{p\times p}\}_{i=1}^{\mathcal{N}^*}$ corresponding to specific out-of-sample diffusion coordinates, $\{\mathbf{\Theta}_i^*\}_{i=1}^{\mathcal{N}}$. Reconstruction of samples is achieved by multiplying the above predicted matrices, such that

$$\mathbf{Y}_i^* = \mathbf{U}_i^* \mathbf{\Sigma}_i^* \mathbf{V}_i^*, \tag{36}$$

where $i = 1, ..., \mathcal{N}_*$ and $\mathbf{Y}_i^* \in \mathbb{R}^{n\times m}$.

4. PREDICTION ACCURACY

To assess the accuracy of predictions of the proposed method, we introduce three metrics. The first scalar metric is the relative L_2 error given by

$$L_2(\mathbf{Y}_{\text{pred}}, \mathbf{Y}_{\text{ref}}) = \frac{\|\mathbf{Y}_{\text{pred}} - \mathbf{Y}_{\text{ref}}\|_2}{\|\mathbf{Y}_{\text{ref}}\|_2}, \tag{37}$$

where $\|\cdot\|_2$ denotes the standard Euclidean norm and $\mathbf{Y}_{\text{pred}}, \mathbf{Y}_{\text{ref}}$ are the prediction and reference responses, respectively. The second scalar metric we introduce is the R^2 score, also known as the coefficient of determination defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{w}\left(\mathbf{Y}_{\text{pred},i} - \mathbf{Y}_{\text{ref},i}\right)^2}{\sum_{i=1}^{w}\left(\mathbf{Y}_{\text{ref},i} - \overline{\mathbf{Y}}_{\text{ref}}\right)^2}, \tag{38}$$

where w is the total number of mesh points of the QoI and $\overline{\mathbf{Y}}_{\text{ref}}$ is the mean reference response. The final metric we introduce is the absolute relative error which measures the error locally, in individual mesh points of the QoIs and is given by the following expression:

$$\epsilon = \left|\frac{\mathbf{Y}_{\text{pred}} - \mathbf{Y}_{\text{ref}}}{\mathbf{Y}_{\text{ref}}}\right|. \tag{39}$$

While the latter nonscalar metric is employed in the sequel to measure the error of individual realizations and provide a visual representation of them, the two scalar metrics are used to evaluate the overall accuracy of the proposed surrogate for a large number of additional realizations.

5. APPLICATIONS

5.1 Application 1: Dielectric Cylinder in Homogeneous Electric Field

In this first example, we consider a model problem from electromagnetic field theory, that of an infinitely long dielectric cylinder suspended in a homogeneous electric field. Due to translational invariance along the z axis the problem can be reduced to two dimensions. Thus, the computational domain is given as $\Omega = [-1, 1] \times [-1, 1]$ and the cylinder's domain by $D_{\text{c}} = \left\{\mathbf{x} = (x, y) \mid \sqrt{x^2 + y^2} \leq r_0\right\}$, where r_0 is the cylinder's radius. The dielectric material of

the cylinder has the relative permittivity $\varepsilon_{\rm c}$, while outside the cylinder's domain the relative permittivity is $\varepsilon_{\rm o}$. The homogeneous electric field is given as $\mathbf{e}_\infty = (E_\infty, 0)$. We further assume Dirichlet bc on the left and right boundaries of the rectangular domain Ω and Neumann bc on its top and bottom boundaries. The Dirichlet and Neumann boundaries are denoted with $\Gamma_{\rm D}$ and $\Gamma_{\rm N}$, respectively. The electric potential $u(\mathbf{x})$ in Ω can be computed by solving the Laplace equation,

$$-\nabla \cdot (\varepsilon(\mathbf{x})\nabla u(\mathbf{x})) = 0, \qquad \mathbf{x} \in \Omega, \tag{40a}$$

$$u(\mathbf{x}) = u^*(\mathbf{x}), \qquad \mathbf{x} \in \Gamma_{\rm D}, \tag{40b}$$

$$(\nabla u(\mathbf{x})) \cdot \mathbf{n} = (\nabla u^*(\mathbf{x})) \cdot \mathbf{n}, \qquad \mathbf{x} \in \Gamma_{\rm N}, \tag{40c}$$

where $\mathbf{n}$ denotes the outer normal unit vector, the permittivity $\varepsilon(\mathbf{x})$ is given as

$$\varepsilon(\mathbf{x}) = \begin{cases} \varepsilon_{\rm c}, & \mathbf{x} \in D_{\rm c}, \\ \varepsilon_{\rm o}, & \mathbf{x} \in \Omega \setminus D_{\rm c}, \end{cases} \tag{41}$$

and u^*, which is also the analytical solution to the problem, is given by

$$u^*(\mathbf{x}) = -E_\infty x \begin{cases} 1 - \dfrac{\varepsilon_{\rm c}/\varepsilon_{\rm o} - 1}{\varepsilon_{\rm c}/\varepsilon_{\rm o} + 1} \dfrac{r_0^2}{x^2 + y^2}, & \mathbf{x} \in \Omega \setminus D_{\rm c}, \\ \dfrac{2}{\varepsilon_{\rm c}/\varepsilon_{\rm o} + 1}, & \mathbf{x} \in D_{\rm c}. \end{cases} \tag{42}$$

We consider variations in the electric potential resulting from stochasticity in two input parameters, the cylinder's radius r_0 and the strength of the electric field E_∞. Information related to parameter description and values is provided in Table 1.

We generate $\mathcal{N} = \{150, 400, 800\}$ training samples $\{\mathscr{X}_i \in \mathbb{R}^{\mathcal{N}\times 2}\}_{i=1}^3$, with corresponding model outputs $\{\mathscr{Y}_i \in \mathbb{R}^{\mathcal{N}\times 6400}\}_{i=1}^3$ where the square computational domain has been discretized in $w = (80 \times 80) = 6400$ mesh points. GDMaps converged to a Grassmann manifold dimension of $p = 30$ which results in matrices on the Grassmannian $\{\mathbf{U}_i, \mathbf{V}_i \in \mathcal{G}_{(30,80)}\}_{i=1}^{\mathcal{N}}$ for each training dataset. Based on the residuals computed by the eigendecomposition of the Markov matrix [55], the first $q = 4$ nontrivial diffusion coordinates are considered, specifically $\{\theta_1, \theta_7, \theta_6, \theta_8\}$, $\{\theta_1, \theta_6, \theta_7, \theta_5\}$, and $\{\theta_1, \theta_7, \theta_5, \theta_8\}$, to represent the embedding structure for the three datasets with $\mathcal{N} = 150$, 400, and 800, respectively. Therefore the method allows us to perform a dimension hyper-reduction from $\mathbb{R}^{6400}$ to $\mathbb{R}^4$, unfold the intrinsic geometric structure of the data, and reveal the essential features. A surrogate model is constructed with a maximum degree of polynomials $s_{\max} = 3$. The adaptive clustering algorithm converged to $\ell = \{13, 25, 30\}$ clusters, respectively. In Fig. 2, we present 2D plots of the diffusion coordinates for $\mathcal{N} = 800$.

A comparison between the reference response and the GDMaps PCE prediction for a random sample ($r_0 = 0.273, E_\infty = 10.523$) is presented in Fig. 3 for all three training datasets. Overall, we observe a very good match between the reference field and surrogate predictions. The relative error is calculated based on Eq. (39), and as expected decreases as the number of training samples increases. To assess more accurately the predictive ability of the surrogate model we compute the relative L_2 error and coefficient of determination (or R^2 score) based on

TABLE 1: Details of the input parameters of the dielectric cylinder application

Parameters		**Uncertainty/value**[**]
Cylinder radius	r_0	$\sim \mathcal{U}(0.20, 0.70)^*$
Strength of electric field	E_∞	$\sim \mathcal{U}(8, 18)^*$
Relative permittivity of cylinder's material	$\varepsilon_{\rm c}$	3
Relative permittivity of surrounding space	$\varepsilon_{\rm o}$	1

[*] $\mathcal{U}(a, b)$ denotes a uniform distribution with lower bound a and upper bound b.
[**] All sizes are expressed in SI units.

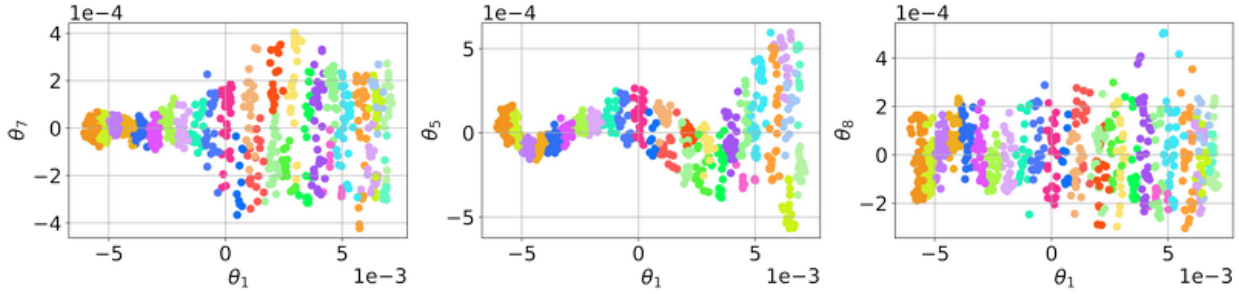

FIG. 2: 2D plots of the diffusion coordinates $\{\theta_1, \theta_7, \theta_5, \theta_8\}$ for Grassmann manifold dimension $p = 30$, $\mathcal{N} = 800$ training samples, and $\ell = 30$ clusters. Different colors denote the different clusters.

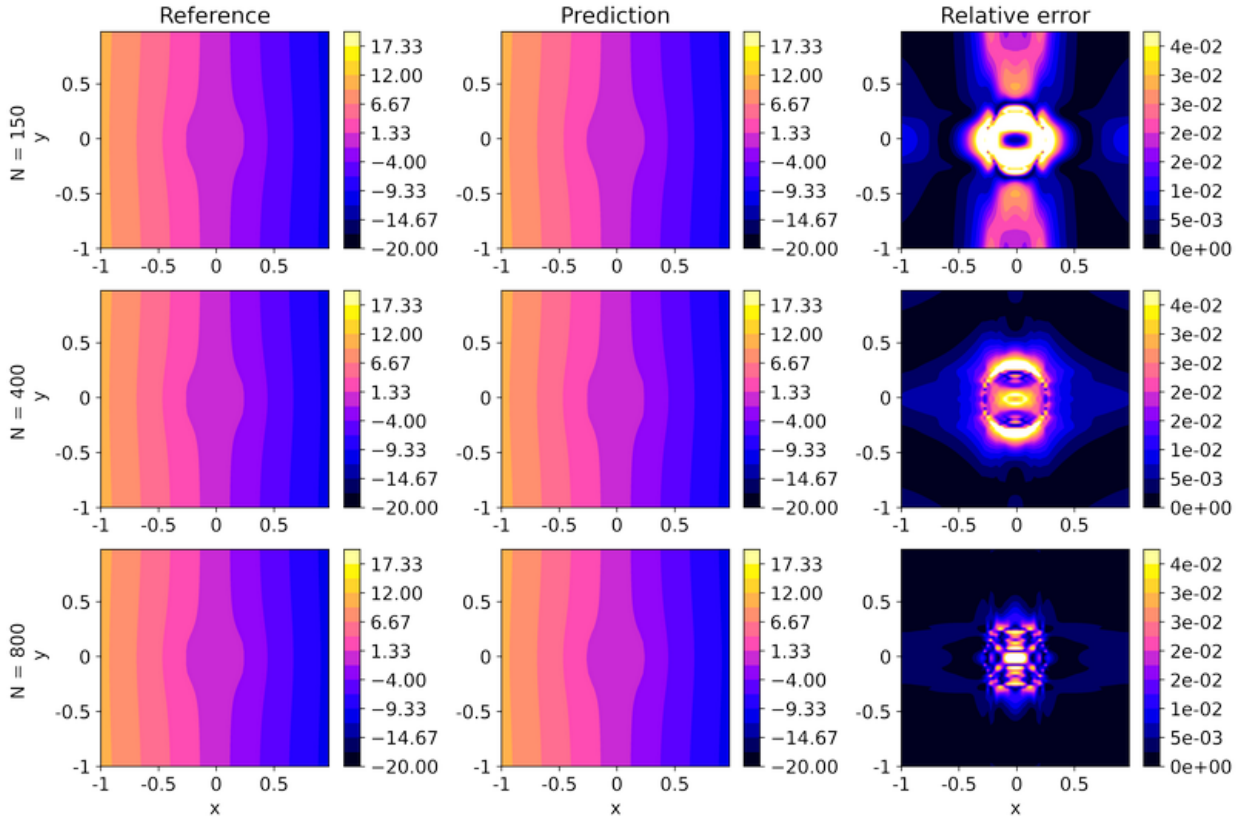

FIG. 3: Reference, prediction, and relative error of the electric potential field $u^*(x, y)$ for the random sample ($r_0 = 0.273$, $E_\infty = 10.523$), $p = 40$, and for $\mathcal{N} = 150$ (first row), $\mathcal{N} = 400$ (second row), and $\mathcal{N} = 800$ (third row) training samples

Eqs. (37) and (38), respectively, for $\mathcal{N}_* = 10{,}000$ testing realizations and we present the first two moments of the corresponding metric value distributions in Table 2. Clearly, there is a significant improvement of results when we increase the number of training samples; however, we observe that the surrogate is able to perform very well in the small-data regime.

In the context of UQ, we next perform moment estimation where we compute the mean field and variance field for $\mathcal{N}_* = 10{,}000$ with Monte Carlo simulation (MCS) on both the original model $\mathcal{M}$ and GDMaps PCE. The results for the moment and variance fields are presented in Figs. 4 and 5, respectively. In both cases, we see a very close agreement between the reference response and the surrogate prediction.

TABLE 2: Relative L_2 error and R^2 score for different training datasets and $\mathcal{N}_* = 10{,}000$ testing realizations

Training data	Relative L_2 error		R^2 score	
	Mean	Std	Mean	Std
$\mathcal{N} = 150$	6.704×10^{-3}	4.071×10^{-3}	9.9993×10^{-1}	8.00×10^{-5}
$\mathcal{N} = 400$	4.967×10^{-3}	2.644×10^{-3}	9.9996×10^{-1}	3.90×10^{-5}
$\mathcal{N} = 800$	4.980×10^{-3}	2.689×10^{-3}	9.9997×10^{-1}	4.00×10^{-5}

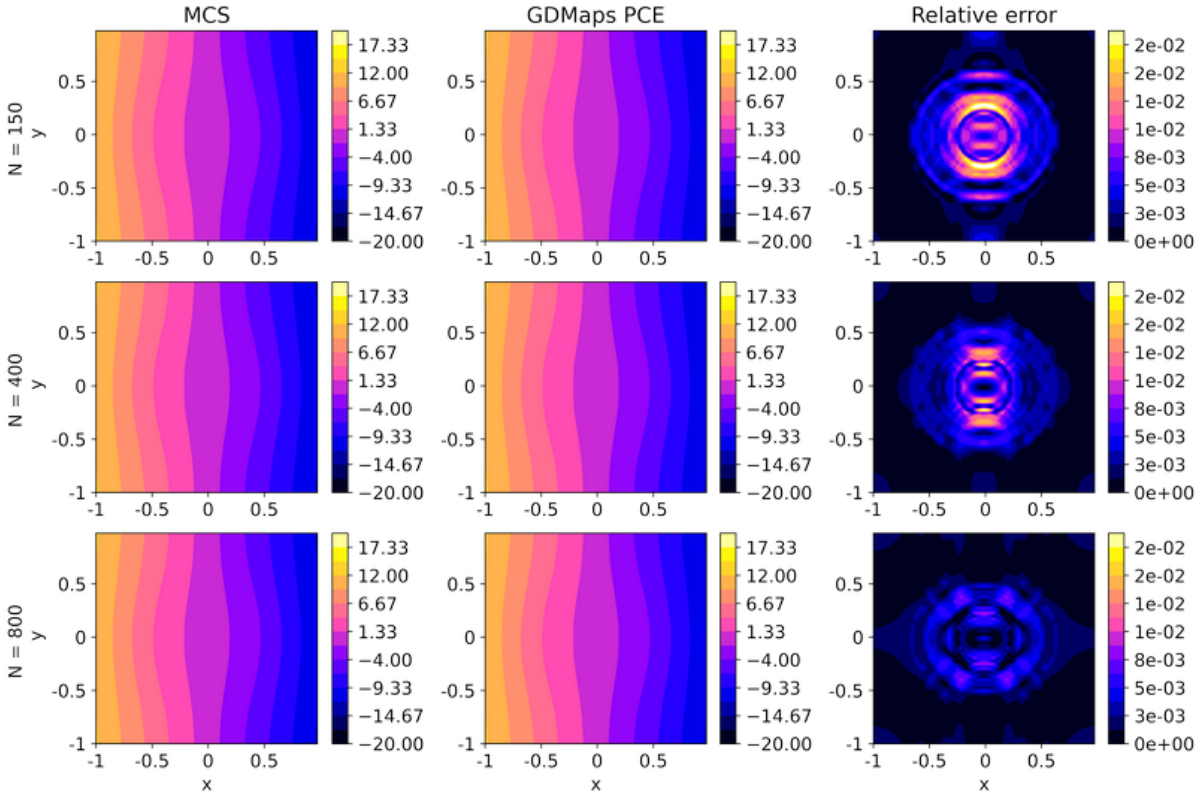

FIG. 4: Mean fields computed with the original model and GDMaps PCE of the electric potential field $u^*(x, y)$ for $\mathcal{N} = 150$ (first row), $\mathcal{N} = 400$ (second row), and $\mathcal{N} = 800$ (third row) training samples and 10,000 testing samples

5.2 Application 2: Lotka-Volterra Dynamical System

In this example, we consider the classic Lotka-Volterra dynamical system [81], also known as the predator-prey equations, an example of a Kolmogorov model which describes the dynamics of a biological system in which two species interact, a predator (e.g., foxes) and a prey (e.g., rabbits). The model is a pair of nonlinear ordinary differential equations (ODEs) defined as follows:

$$\begin{aligned} \frac{du}{dt} &= \alpha u - \beta uv, \\ \frac{dv}{dt} &= \delta uv - \gamma v, \end{aligned} \tag{43}$$

where u is the prey population, v is the predator population, and $\alpha, \beta, \gamma, \delta$ are stochastic model parameters described in Table 3. The equations have periodic solutions with 90° phase difference and a linearization leads to solutions similar to those of a simple harmonic oscillator.

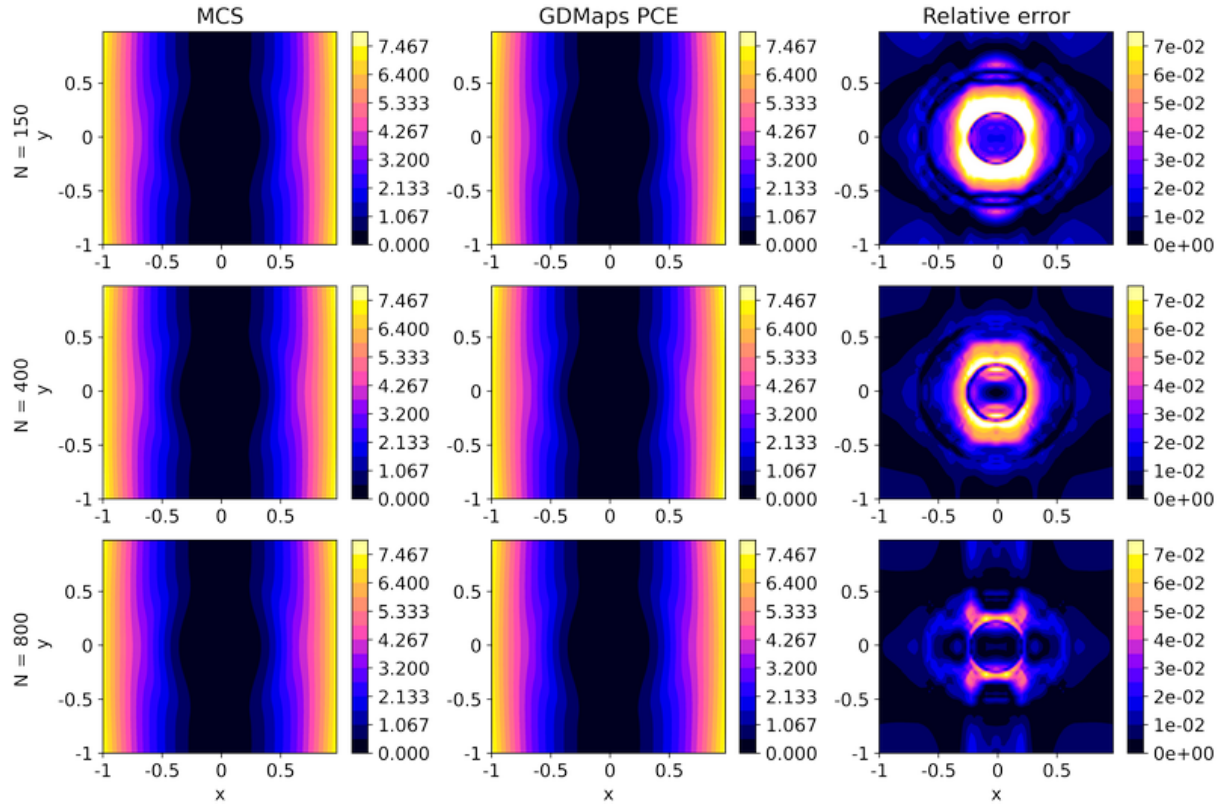

FIG. 5: Variance fields computed with the original model and GDMaps PCE of the electric potential field $u^*(x, y)$ for $\mathcal{N} = 150$ (first row), $\mathcal{N} = 400$ (second row), and $\mathcal{N} = 800$ (third row) training samples and 10,000 testing samples

TABLE 3: Details of the state variables and input parameters of the Lotka-Volterra equations

Description of variables/parameters		Uncertainty/value
Population of prey species	u	$u(t=0) = 10$
Population of predator species	v	$v(t=0) = 5$
Natural growing rate of preys when no predator exists	α	$\sim \mathcal{U}(0.90, 1)$
Natural dying rate of preys due to predation	β	$\sim \mathcal{U}(0.10, 0.15)$
Natural dying rate of predator when no prey exists	γ	1.50
Reproduction rate of predators per prey eaten	δ	0.75

For illustration, we consider two stochastic parameters α and β and employ GDMaps PCE to construct a surrogate model to predict the trajectory of both predator and prey species over time. We generate $\mathcal{N} = \{50, 150, 600\}$ training samples $\{\mathscr{X}_i \in \mathbb{R}^{\mathcal{N}_i \times 2}\}_{i=1}^{3}$. For each training dataset, the system is solved using a fourth-order Runge-Kutta method with period $T = 25$, discretized in $w = 512$ points, thus resulting in a response matrix $\mathscr{Y} \in \mathbb{R}^{\mathcal{N} \times 1024}$ where the corresponding solutions $\{u(t), v(t)\}$ for each sample are concatenated in a single vector. Each solutions is reshaped to a square matrix $\{\mathbf{Y}_i \in \mathbb{R}^{32 \times 32}\}_{i=1}^{\mathcal{N}}$ and GDMaps is performed for a constant value of $p = 10$, which results in matrices on the Grassmannian $\{\mathbf{U}_i, \mathbf{V}_i \in \mathcal{G}_{(10,32)}\}_{i=1}^{\mathcal{N}}$. By keeping $q = 3$ parsimoniously selected diffusion coordinates we converged to the first non-trivial coordinates $\mathbf{\Theta}_i = \{\theta_1, \theta_2, \theta_5\}$, where $\{\mathbf{\Theta}_i \in \mathbb{R}^3\}_{i=1}^{\mathcal{N}}$. Finally, the PCE surrogate is constructed with a maximum polynomial degree $s_{\max} = 3$.

Results from the adaptive clustering algorithm for different training dataset sizes are presented in Fig. 6. The algorithm converged to $\ell = \{4, 13, 44\}$ clusters, respectively. In Fig. 7, the embedding represented by the diffusion

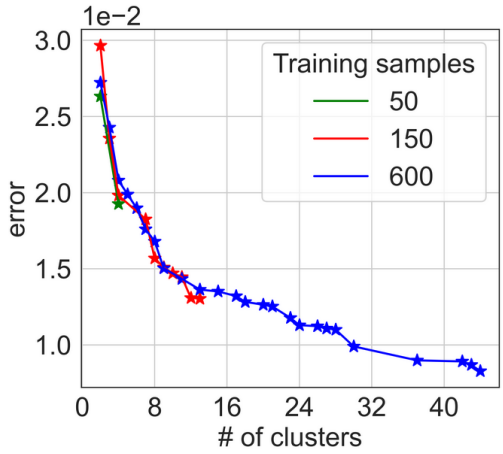

FIG. 6: Average mean square error (MSEs) associated with the projection of clusters of points $\{\mathbf{U}_i, \mathbf{V}_i\}_{i=1}^{\mathcal{N}_k}$ to the tangent space of the Grassmannian $\mathcal{G}_{(10,32)}$ and back as a function of the number of clusters for $\mathcal{N} = \{50, 150, 600\}$

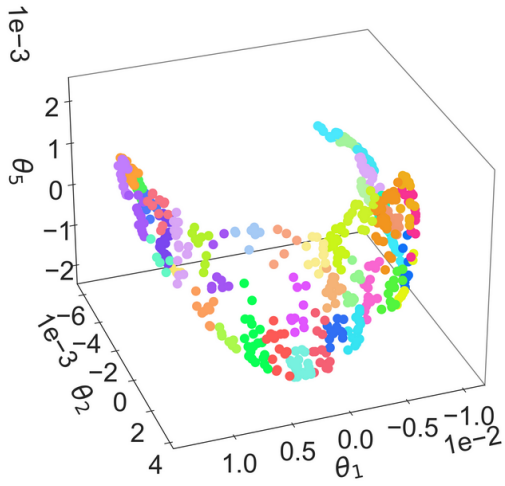

FIG. 7: Clusters of points on the Grassmannian diffusion manifold represented by the first three nontrivial diffusion coordinates $\{\theta_1, \theta_2, \theta_5\}$ for $\mathcal{N} = 600$ and $\ell = 44$

coordinates for $\mathcal{N} = 600$ where $\ell = 44$ is also shown. The prediction of the surrogate for the trajectories of both the prey and predator species is presented in Fig. 8 where we compare reference solutions and predictions for three random samples. We observe a considerable improvement of results when the number of training data increases, but nonetheless find good agreement between some points even for small training set sizes. Finally, we compare $\mathcal{N}_* = 5000$ testing realizations of the surrogate model with the corresponding reference solutions and plot the corresponding error distributions in Fig. 9. We observe a rapid reduction in error (lower L_2, higher R^2) from 50 to 150 training data with continued, but less significant improvement from 150 to 600 training data.

5.3 Application 3: Advection-Diffusion-Reaction Equations

In the third example, we consider a system of *advection-diffusion-reaction* equations modeling a first-order chemical reaction between two species A and B; the result is a formed species C in some domain Ω. The reaction reads

$$A + B \xrightarrow{\epsilon} C, \tag{44a}$$

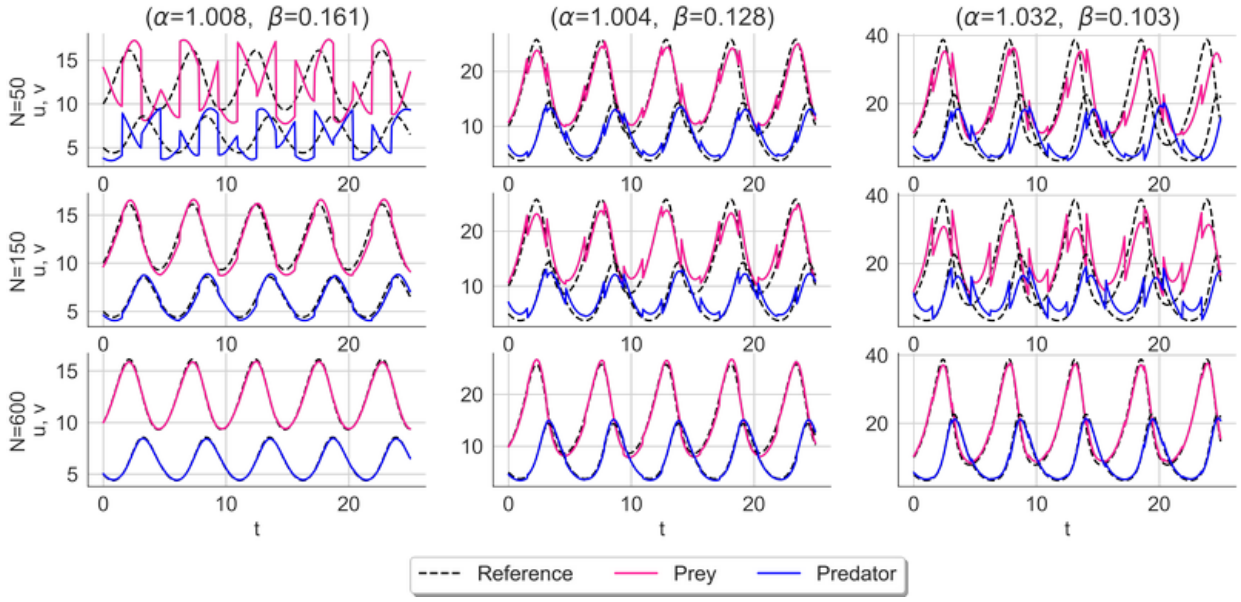

FIG. 8: Comparison of trajectories of reference solutions (dashed) and predictions for the prey (solid magenta) and predator (solid blue) species for $\mathcal{N} = \{50, 150, 600\}$ (rows) and three randomly generated samples (columns)

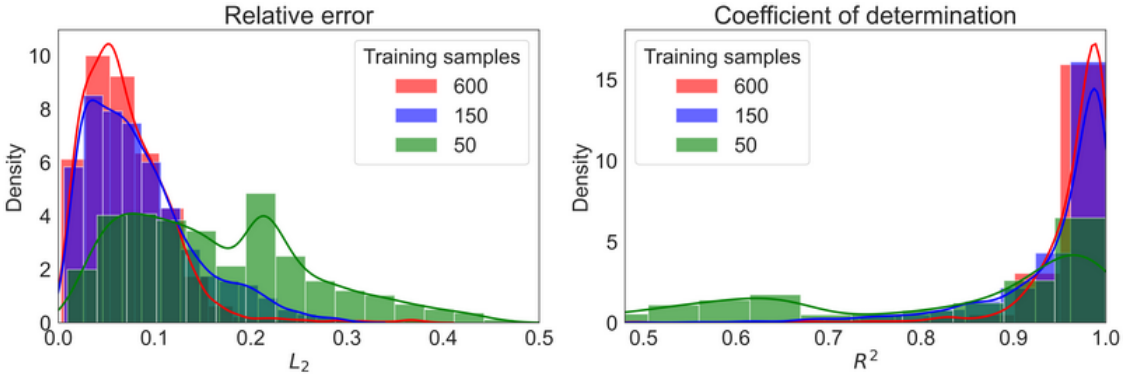

FIG. 9: Histograms of the relative L_2 error (left) and the coefficient of determination or R^2 score (right) of predictions and reference solutions for $\mathcal{N} = \{50, 150, 600\}$ and $\mathcal{N}_* = 5000$ testing realizations

$$\frac{d[C]}{dt} = K[A][B], \tag{44b}$$

where $[A], [B], [C]$ are the concentrations of the three species, ϵ is the diffusion coefficient, and K is the reaction rate. According to the mass action law of chemical kinetics, the reaction rate of C is proportional to the concentration of the two species A, B [82]. The chemical reaction is modeled by the following set of equations:

$$\frac{\partial[A]}{\partial t} + v \cdot \nabla[A] - \nabla \cdot (\epsilon \nabla[A]) = f_A - K[A][B], \tag{45a}$$

$$\frac{\partial[B]}{\partial t} + v \cdot \nabla[B] - \nabla \cdot (\epsilon \nabla[B]) = f_B - K[A][B], \tag{45b}$$

$$\frac{\partial[C]}{\partial t} + v \cdot \nabla[C] - \nabla \cdot (\epsilon \nabla[C]) = f_C + K[A][B] - K[C], \tag{45c}$$

where species A, B, and C diffuse throughout Ω (third terms in the left-hand side) and are advected with velocity v (second terms in the left-hand side). The chemical reaction is represented in the right-hand side of Eq. (45) with source terms f_A, f_B, and f_C for species A, B, and C, respectively.

The chemical reaction takes place in a velocity field flowing around a cylinder and thus the above equations are coupled with the incompressible Navier-Stokes nonlinear PDEs, defined as

$$\rho\Big(\frac{\partial v}{\partial t} + v \cdot \nabla v\Big) = \nabla \cdot \sigma(v,p) + f, \tag{46a}$$

$$\nabla \cdot v = 0. \tag{46b}$$

Species A, B are injected into the system from two points at the top and bottom of the cylinder via the nonzero source terms f_A, f_B and then advect and diffuse through the system. The third source term is set to $f_C = 0$ and species C is formed only as the result of the reaction of A and B. For the numerical implementation of the above coupled system of equations we use the FEniCS package [83].

We assume that stochasticity in the above system of coupled PDEs results from variations in the reaction rate K and the diffusion coefficient ϵ. In Table 4, the initial conditions and the distributions of stochastic parameters are presented. As QoI we consider the concentration of species C at the final time step, where $T = 5$ is the total simulation time and $n = 5000$ is the number of time steps. The simulation takes place in a rectangular domain $\Omega = [0, 2.2] \times [0, 0.41]$ while the cylinder is centered at $c = (0.2, 0.2)$ with radius $r = 0.05$. The domain is discretized with $w = 2304$ mesh points. We generate $\mathcal{N} = 600$ training samples $\mathbf{X}_i \in \mathbb{R}^2, i = 1, \ldots, \mathcal{N}$, and corresponding model responses $\mathbf{Y}_i \in \mathbb{R}^{2304}, i = 1, \ldots, \mathcal{N}$. In Fig. 10, nine realizations of the stochastic field solution are shown. For the parametric uncertainty considered, we observe significant variations between the various profiles representing the concentration of C.

The solutions are reshaped to square matrices $\{\mathbf{Y}_i \in \mathbb{R}^{48\times 48}\}_{i=1}^{\mathcal{N}}$ and GDMaps is performed. In this example, we aim to explore the method's predictive ability by varying the dimension of the Grassmannian on which the data

TABLE 4: Details of the state variables and input parameters of the system of advection-diffusion-reaction equations

Description of variables/parameters		**Uncertainty/value**
Concentration of species A	$[A]$	$[A](t=0) = 0$
Concentration of species B	$[B]$	$[B](t=0) = 0$
Concentration of species C	$[C]$	$[C](t=0) = 0$
Reaction rate	K	$\sim \mathcal{U}(7.0, 13.0)$
Diffusion coefficient	ϵ	$\sim \mathcal{U}(0.005, 0.015)$

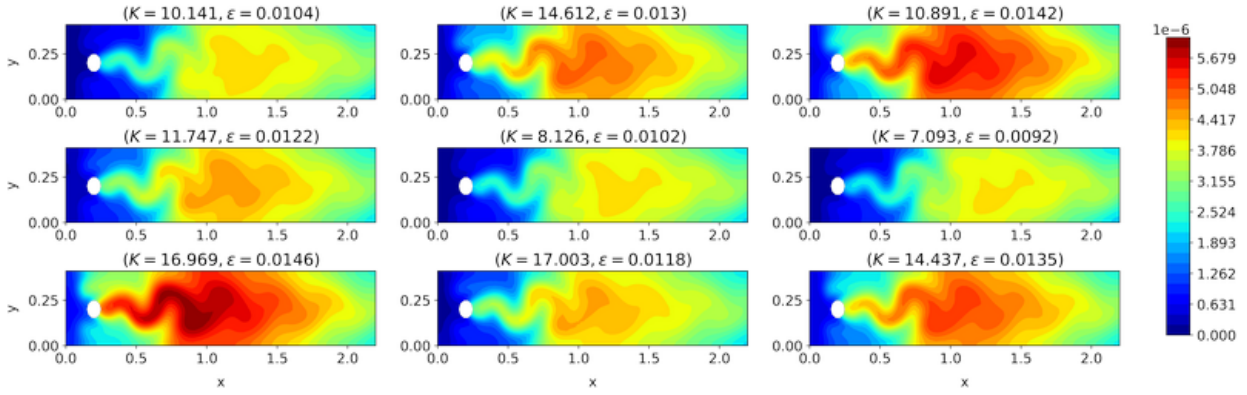

FIG. 10: Realizations of the stochastic field $\mathbf{Y}$, representing the concentration of species C at the final time step, as a result of the chemical reaction of A and B with associated stochastic parameter values $\mathscr{X}$

are projected. We consider three values, $p = \{16, 28, 40\}$ which result in matrices on the Grassmannian $\{\mathbf{U}_i, \mathbf{V}_i \in \mathcal{G}_{(p,48)}\}_{i=1}^{\mathcal{N}}$. Based on the decay of eigenvalues resulting from the DMaps we retain $q = 3$ of diffusion coordinates, resulting in $\{\mathbf{\Theta}_i \in \mathbb{R}^3\}_{i=1}^{\mathcal{N}}$. The PCE surrogate is constructed with a maximum polynomial degree, $s_{\max} = 3$. The adaptive clustering algorithm resulted in $\ell = \{37, 31, 38\}$ clusters, respectively.

In Fig. 11, we present 3D plots of the parsimoniously selected diffusion coordinates on the Grassmannian diffusion manifold for all three cases. In Fig. 12, we present the surrogate model predictions for a random sample $(K = 16.138, \epsilon = 0.00127)$ for $p = \{16, 28, 40\}$, where the reference response, the surrogate prediction, and the relative error computed with Eq. (39) are shown. We observe that for a small dimension of the Grassmann manifold, the surrogate is able to capture the local intensities of the concentration of species C, however, with significant noise in the prediction. As we increase the dimension, we notice a significant improvement in the results and a reduction of the relative error. We observe that even though the number of diffusion coordinates remains constant in all three cases (i.e., $q = 3$), the dimension of the Grassmann manifold on which the data are projected in the intermediate dimension reduction step significantly affects the predictive ability of the surrogate. We note that the larger relative errors near the inflow boundary are caused due to the concentration (denominator) being close to zero.

To assess the overall performance of the surrogate, in Fig. 13 we plot the distributions of the relative L_2 error and the R^2 score for 1500 test realizations. Clearly, the two distributions are getting closer to zero and 1, respectively, as the dimensionallity of the Grassmannian increases. Finally, for the same test realizations we perform moment estimation and calculate the mean and variance fields of the concentration of species C via MCS with the original model and with GDMaps PCE for the same testing realizations and for $p = 40$, shown in Fig. 14. As observed from the plots, the method is able to accurately predict the first two moments of the field.

We have demonstrated that GDMaps PCE performs very well in cases where large variability of model solutions is considered. The proposed approach results in significant cost reductions. More specifically, while a forward model

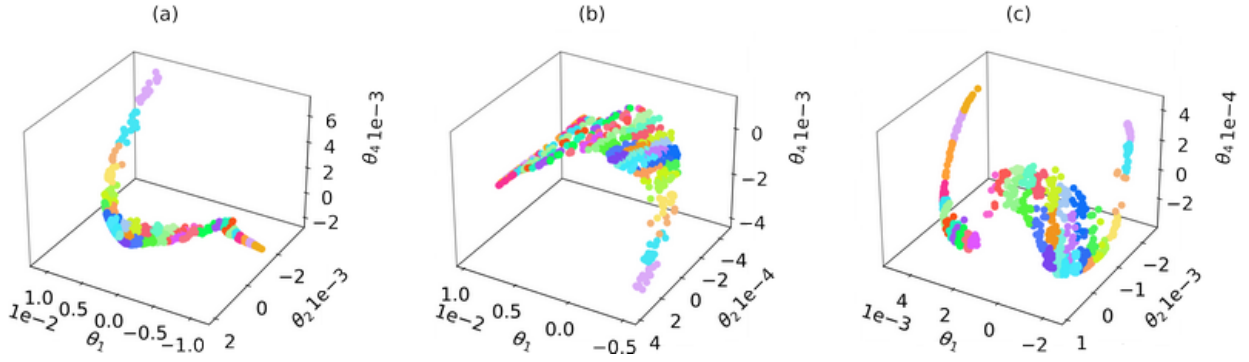

FIG. 11: Clustered diffusion coordinates $\{\theta_1, \theta_2, \theta_4\}$ for Grassmann manifold dimension of (a) $p = 16$, (b) $p = 28$, and (c) $p = 40$ for $\mathcal{N} = 600$ training samples

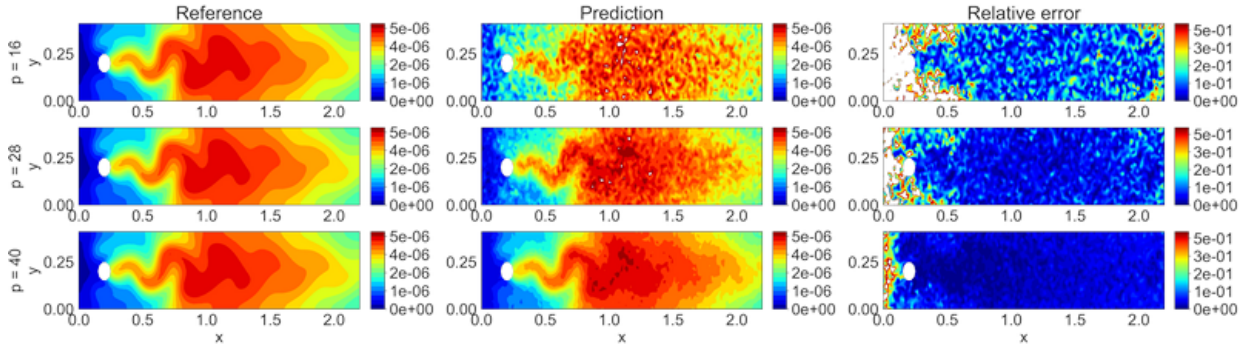

FIG. 12: Reference solution, surrogate prediction, and relative error of the concentration of species C for parameter realizations $(K = 16.138, \epsilon = 0.00127)$ and Grassmann manifold dimension $p = \{16, 28, 40\}$

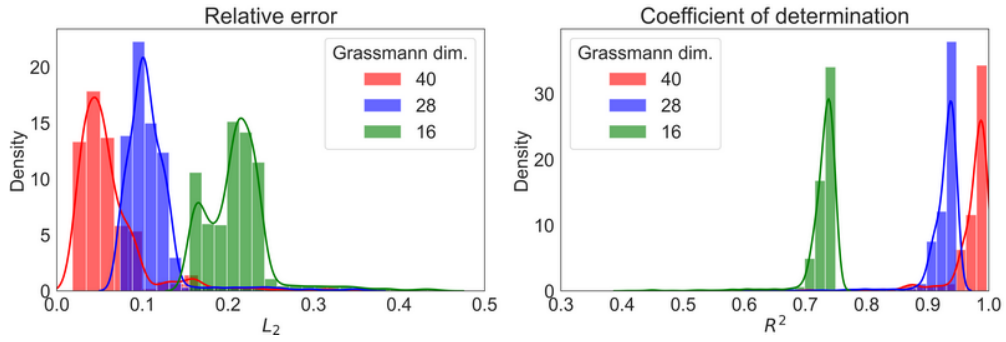

FIG. 13: Histograms of the relative L_2 error (left) and the coefficient of determination or R^2 score (right) of predictions and reference solutions for $\mathcal{N}_* = 1500$ testing realizations, and Grassmann manifold dimension of (a) $p = 16$, (b) $p = 28$, and (c) $p = 40$

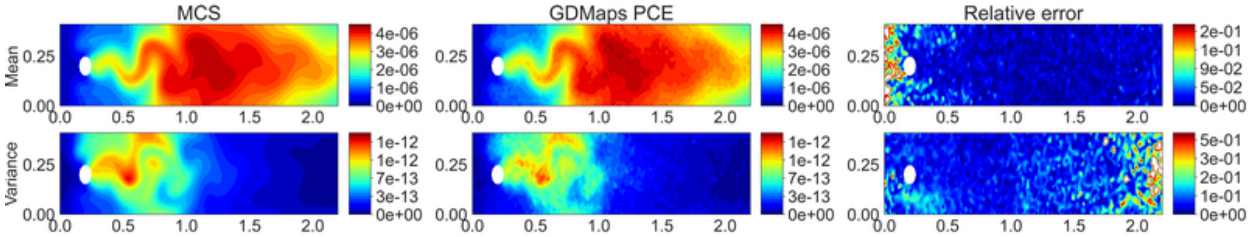

FIG. 14: Mean field (top row) and variance field (bottom row) for Monte Carlo simulation of $\mathcal{N}_* = 1500$ samples, surrogate prediction, and relative error for $\mathcal{N} = 600$ training samples and Grassmann dimension $p = 40$

evaluation requires approximately 45 sec to complete, the proposed surrogate is able to predict model responses in an average of 0.093 sec, i.e., 483× faster. Extrapolating these times, we see that a MC simulation of 10,000 samples with the original model would require $\sim$ 5 days of CPU time to complete, while a MC simulation with GDMaps PCE would only need $\sim$ 15 min of CPU time. The computational gains of the proposed framework become all the more prominent as the complexity, output dimensionality, and, therefore, the cost of the model increases.

6. DISCUSSION AND CONCLUSIONS

This paper introduces a manifold learning-based approach for the construction of surrogate models on lower-dimensional manifolds for UQ in complex high-dimensional systems. The GDMaps PCE framework is specially designed for applications of high-output dimensionality and nonlinearity. We introduced an encoder-decoder type framework in which GDMaps, a two-step dimension reduction technique for feature extraction, is performed to project data onto a Grassmannian and consequently onto a diffusion manifold. Diffusion coordinates are used to represent a lower-dimensional embedding capable of capturing the salient information of the empirical dataset. A PCE surrogate is constructed on the latent space, and an adaptive clustering technique is proposed to identify regions of response similarity and consequently construct local geometric harmonics to naturally perform out-of-sample predictions.

We explored the method's capabilities and limitations on three applications from electromagnetic field theory, nonlinear dynamics, and chemical kinetics. Numerical results demonstrate that the proposed approach is able to accurately predict new out-of-sample solutions. The dimension of the Grassmann manifold is an important factor for the GDMaps PCE method. However, in all studied applications, we demonstrated that a very small number of coordinates (two to four) representing the diffusion manifold can sufficiently capture the essential features. Furthermore, the method performed successfully under the use of small datasets and it resulted in significant reductions of the computational cost associated with the high-fidelity simulations. In the context of UQ, we have shown that GDMaps PCE provides an appropriate framework to perform statistical moment estimation in a computationally efficient manner and enables Monte Carlo simulations, which would otherwise be prohibitively expensive, to compute with the

original model. An interesting future direction would be to explore whether PCE coefficients computed with data on the latent space could provide useful information of QoIs in the ambient space for direct moment estimation and sensitivity analysis.

Finally, although the proposed surrogate modeling method is ideally applicable to cases of complex models generating high-dimensional responses, its computational cost will still become intractable in cases where the input parameter space is also characterized by high-dimensionality. In such cases, methods exploiting sparse representations of PCE surrogates should be considered. These cases have not been considered. In addition, depending on the variability of the data on the diffusion manifold, an experimental design based on a standard random or quasirandom sampling technique might not be ideal. Greedy sampling techniques can be considered for such cases. These limitations form challenges to be addressed in future work.

REFERENCES

1. Bhosekar, A. and Ierapetritou, M., Advances in Surrogate Based Modeling, Feasibility Analysis, and Optimization: A Review, *Comput. Chem. Eng.*, **108**:250–267, 2018.
2. Xiu, D. and Karniadakis, G.E., The Wiener-Askey Polynomial Chaos for Stochastic Differential Equations, *SIAM J. Sci. Comput.*, **24**(2):619–644, 2002.
3. Zhou, Y., Lu, Z., Cheng, K., and Shi, Y., An Expanded Sparse Bayesian Learning Method for Polynomial Chaos Expansion, *Mech. Syst. Signal Proces.*, **128**:153–171, 2019.
4. Hadigol, M. and Doostan, A., Least Squares Polynomial Chaos Expansion: A Review of Sampling Strategies, *Comput. Methods Appl. Mech. Eng.*, **332**:382–407, 2018.
5. Babuška, I., Nobile, F., and Tempone, R., A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data, *SIAM J. Numer. Anal.*, **45**(3):1005–1034, 2007.
6. Sudret, B., Global Sensitivity Analysis Using Polynomial Chaos Expansions, *Reliab. Eng. Syst. Saf.*, **93**(7):964–979, 2008.
7. Shao, Q., Younes, A., Fahs, M., and Mara, T.A., Bayesian Sparse Polynomial Chaos Expansion for Global Sensitivity Analysis, *Comput. Methods Appl. Mech. Eng.*, **318**:474–496, 2017.
8. Crestaux, T., Le Maıtre, O., and Martinez, J.M., Polynomial Chaos Expansion for Sensitivity Analysis, *Reliab. Eng. Syst. Saf.*, **94**(7):1161–1172, 2009.
9. Lüthen, N., Marelli, S., and Sudret, B., Sparse Polynomial Chaos Expansions: Literature Survey and Benchmark, *Math. Numer. Anal.*, arXiv:2002.01290, 2020.
10. Blatman, G. and Sudret, B., Adaptive Sparse Polynomial Chaos Expansion Based on Least Angle Regression, *J. Comput. Phys.*, **230**(6):2345–2367, 2011.
11. Alemazkoor, N. and Meidani, H., A Preconditioning Approach for Improved Estimation of Sparse Polynomial Chaos Expansions, *Comput. Methods Appl. Mech. Eng.*, **342**:474–489, 2018.
12. Jakeman, J.D., Eldred, M.S., and Sargsyan, K., Enhancing ℓ_1-Minimization Estimates of Polynomial Chaos Expansions Using Basis Selection, *J. Comput. Phys.*, **289**:18–34, 2015.
13. Hampton, J. and Doostan, A., Basis Adaptive Sample Efficient Polynomial Chaos (BASE-PC), *J. Comput. Phys.*, **371**:20–49, 2018.
14. Loukrezis, D., Galetzka, A., and De Gersem, H., Robust Adaptive Least Squares Polynomial Chaos Expansions in High-Frequency Applications, *Int. J. Numer. Modell. Electron. Networks, Devices Fields*, **33**(6):e2725, 2020.
15. Hombal, V. and Mahadevan, S., Surrogate Modeling of 3D Crack Growth, *Int. J. Fatigue*, **47**:90–99, 2013.
16. Boukouvala, F., Gao, Y., Muzzio, F., and Ierapetritou, M.G., Reduced-Order Discrete Element Method Modeling, *Chem. Eng. Sci.*, **95**:12–26, 2013.
17. Amsallem, D. and Farhat, C., Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity, *AIAA J.*, **46**(7):1803–1813, 2008.
18. Zimmermann, R., Gradient-Enhanced Surrogate Modeling Based on Proper Orthogonal Decomposition, *J. Comput. Appl. Math.*, **237**(1):403–418, 2013.
19. Carlberg, K. and Farhat, C., A Low-Cost, Goal-Oriented Compact Proper Orthogonal Decomposition Basis for Model Reduction of Static Systems, *Int. J. Numer. Methods Eng.*, **86**(3):381–402, 2011.

20. Nath, P., Hu, Z., and Mahadevan, S., Sensor Placement for Calibration of Spatially Varying Model Parameters, *J. Comput. Phys.*, **343**:150–169, 2017.

21. Tripathy, R., Bilionis, I., and Gonzalez, M., Gaussian Processes with Built-In Dimensionality Reduction: Applications to High-Dimensional Uncertainty Propagation, *J. Comput. Phys.*, **321**:191–223, 2016.

22. Vohra, M., Nath, P., Mahadevan, S., and Lee, Y.T.T., Fast Surrogate Modeling Using Dimensionality Reduction in Model Inputs and Field Output: Application to Additive Manufacturing, *Reliab. Eng. Syst. Saf.*, **201**:106986, 2020.

23. Constantine, P.G., Dow, E., and Wang, Q., Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces, *SIAM J. Sci. Comput.*, **36**(4):A1500–A1524, 2014.

24. Constantine, P.G., *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, Philadelphia: SIAM, 2015.

25. Bigoni, D., Marzouk, Y., Prieur, C., and Zahm, O., Nonlinear Dimension Reduction for Surrogate Modeling Using Gradient Information, *Math. Numer. Anal.*, arXiv:2102.10351, 2021.

26. Ji, W., Wang, J., Zahm, O., Marzouk, Y.M., Yang, B., Ren, Z., and Law, C.K., Shared Low-Dimensional Subspaces for Propagating Kinetic Uncertainty to Multiple Outputs, *Combust. Flame*, **190**:146–157, 2018.

27. Zahm, O., Constantine, P.G., Prieur, C., and Marzouk, Y.M., Gradient-Based Dimension Reduction of Multivariate Vector-Valued Functions, *SIAM J. Sci. Comput.*, **42**(1):A534–A558, 2020.

28. Giovanis, D.G. and Shields, M.D., Uncertainty Quantification for Complex Systems with Very High Dimensional Response Using Grassmann Manifold Variations, *J. Comput. Phys.*, **364**:393–415, 2018.

29. Giovanis, D.G. and Shields, M.D., Data-Driven Surrogates for High Dimensional Models Using Gaussian Process Regression on the Grassmann Manifold, *Comput. Methods Appl. Mech. Eng.*, **370**:113269, 2020.

30. Kontolati, K., Alix-Williams, D., Boffi, N.M., Falk, M.L., Rycroft, C.H., and Shields, M.D., Manifold Learning for Coarse-Graining Atomistic Simulations: Application to Amorphous Solids, *Acta Mater.*, **215**:117008, 2021.

31. Coifman, R.R. and Lafon, S., Diffusion Maps, *Appl. Comput. Harmonic Anal.*, **21**(1):5–30, 2006.

32. Soize, C. and Ghanem, R., Data-Driven Probability Concentration and Sampling on Manifold, *J. Comput. Phys.*, **321**:242–258, 2016.

33. Soize, C. and Ghanem, R., Polynomial Chaos Representation of Databases on Manifolds, *J. Comput. Phys.*, **335**:201–221, 2017.

34. Soize, C. and Ghanem, R., Probabilistic Learning on Manifolds Constrained by Nonlinear Partial Differential Equations for Small Datasets, *Comput. Methods Appl. Mech. Eng.*, **380**:113777, 2021.

35. Kalogeris, I. and Papadopoulos, V., Diffusion Maps-Based Surrogate Modeling: An Alternative Machine Learning Approach, *Int. J. Numer. Methods Eng.*, **121**(4):602–620, 2020.

36. Koronaki, E., Nikas, A., and Boudouvis, A., A Data-Driven Reduced-Order Model of Nonlinear Processes Based on Diffusion Maps and Artificial Neural Networks, *Chem. Eng. J.*, **397**:125475, 2020.

37. Lataniotis, C., Marelli, S., and Sudret, B., Extending Classical Surrogate Modeling to High Dimensions through Supervised Dimensionality Reduction: A Data-Driven Approach, *Int. J. Uncertainty Quantification*, **10**(1):55–82, 2020.

38. Schmidhuber, J., Deep Learning in Neural Networks: An Overview, *Neural Networks*, **61**:85–117, 2015.

39. Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y., *Deep Learning*, Vol. 1, Cambridge, MA: The MIT Press, 2016.

40. Wang, Y., Yao, H., and Zhao, S., Auto-Encoder Based Dimensionality Reduction, *Neurocomputing*, **184**:232–242, 2016.

41. Rawat, W. and Wang, Z., Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review, *Neural Comput.*, **29**(9):2352–2449, 2017.

42. Tripathy, R.K. and Bilionis, I., Deep UQ: Learning Deep Neural Network Surrogate Models for High Dimensional Uncertainty Quantification, *J. Comput. Phys.*, **375**:565–588, 2018.

43. Nikolopoulos, S., Kalogeris, I., and Papadopoulos, V., Non-Intrusive Surrogate Modeling for Parametrized Time-Dependent PDEs Using Convolutional Autoencoders, *Math. Numer. Anal.*, arXiv:2101.05555, 2021.

44. Zhu, Y. and Zabaras, N., Bayesian Deep Convolutional Encoder–Decoder Networks for Surrogate Modeling and Uncertainty Quantification, *J. Comput. Phys.*, **366**:415–447, 2018.

45. Mo, S., Zhu, Y., Zabaras, N., Shi, X., and Wu, J., Deep Convolutional Encoder-Decoder Networks for Uncertainty Quantification of Dynamic Multiphase Flow in Heterogeneous Media, *Water Resour. Res.*, **55**(1):703–728, 2019.

46. Thuerey, N., Weißenow, K., Prantl, L., and Hu, X., Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows, *AIAA J.*, **58**(1):25–36, 2020.
47. Wang, N., Chang, H., and Zhang, D., Efficient Uncertainty Quantification for Dynamic Subsurface Flow with Surrogate by Theory-Guided Neural Network, *Comput. Methods Appl. Mech. Eng.*, **373**:113492, 2021.
48. Mo, S., Zabaras, N., Shi, X., and Wu, J., Deep Autoregressive Neural Networks for High-Dimensional Inverse Problems in Groundwater Contaminant Source Identification, *Water Resour. Res.*, **55**(5):3856–3881, 2019.
49. Hesthaven, J.S. and Ubbiali, S., Non-Intrusive Reduced Order Modeling of Nonlinear Problems Using Neural Networks, *J. Comput. Phys.*, **363**:55–78, 2018.
50. dos Santos, K.R., Giovanis, D.G., and Shields, M.D., Grassmannian Diffusion Maps Based Dimension Reduction and Classification for High-Dimensional Data, *Comput. Sci. Mach. Learn.*, arXiv:2009.07547, 2020.
51. Zhang, J., Zhu, G., Heath, R.W., Jr., and Huang, K., Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning, *Comput. Sci. Mach. Learn.*, arXiv:1808.02229, 2018.
52. Edelman, A., Arias, T.A., and Smith, S.T., The Geometry of Algorithms with Orthogonality Constraints, *SIAM J. Matrix Anal. Appl.*, **20**(2):303–353, 1998.
53. Coifman, R.R. and Lafon, S., Geometric Harmonics: A Novel Tool for Multiscale Out-Of-Sample Extension of Empirical Functions, *Appl. Comput. Harmonic Anal.*, **21**(1):31–52, 2006.
54. Olivier, A., Giovanis, D., Aakash, B., Chauhan, M., Vandanapu, L., and Shields, M.D., UQpy: A General Purpose Python Package and Development Environment for Uncertainty Quantification, *J. Comput. Sci.*, **47**:101204, 2020.
55. Dsilva, C.J., Talmon, R., Coifman, R.R., and Kevrekidis, I.G., Parsimonious Representation of Nonlinear Dynamical Systems through Manifold Learning: A Chemotaxis Case Study, *Appl. Comput. Harmonic Anal.*, **44**(3):759–773, 2018.
56. Absil, P.A., Mahony, R., and Sepulchre, R., Riemannian Geometry of Grassmann Manifolds with a View on Algorithmic Computation, *Acta Appl. Math.*, **80**(2):199–220, 2004.
57. Ye, K. and Lim, L.H., Schubert Varieties and Distances between Subspaces of Different Dimensions, *SIAM J. Matrix Anal. Appl.*, **37**(3):1176–1197, 2016.
58. Ye, K., Wong, K.S.W., and Lim, L.H., Optimization on Flag Manifolds, *Math. Optim. Control*, arXiv:1907.00949, 2019.
59. Begelfor, E. and Werman, M., Affine Invariance Revisited, in *2006 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'06)*, IEEE, Vol. 2, pp. 2087–2094, 2006.
60. Harandi, M.T., Salzmann, M., Jayasumana, S., Hartley, R., and Li, H., Expanding the Family of Grassmannian Kernels: An Embedding Perspective, in *European Conference on Computer Vision*, Berlin: Springer, pp. 408–423, 2014.
61. Hamm, J. and Lee, D., Extended Grassmann Kernels for Subspace-Based Learning, *Adv. Neural Inf. Proces. Syst.*, **21**:601–608, 2008.
62. Thiem, T.N., Kooshkbaghi, M., Bertalan, T., Laing, C.R., and Kevrekidis, I.G., Emergent Spaces for Coupled Oscillators, *Front. Comput. Neurosci.*, **14**:36, 2020.
63. Feinberg, J., Eck, V.G., and Langtangen, H.P., Multivariate Polynomial Chaos Expansions with Dependent Variables, *SIAM J. Sci. Comput.*, **40**(1):A199–A223, 2018.
64. Jakeman, J.D., Franzelin, F., Narayan, A., Eldred, M., and Plfüger, D., Polynomial Chaos Expansions for Dependent Random Variables, *Comput. Methods Appl. Mech. Eng.*, **351**:643–666, 2019.
65. Rahman, S., A Polynomial Chaos Expansion in Dependent Random Variables, *J. Math. Anal. Appl.*, **464**(1):749–775, 2018.
66. Bobrowski, A., *Functional Analysis for Probability and Stochastic Processes: An Introduction*, Cambridge: Cambridge University Press, 2005.
67. Wan, X. and Karniadakis, G.E., Multi-Element Generalized Polynomial Chaos for Arbitrary Probability Measures, *SIAM J. Sci. Comput.*, **28**(3):901–928, 2006.
68. Soize, C. and Ghanem, R., Physical Systems with Random Uncertainties: Chaos Representations with Arbitrary Probability Measure, *SIAM J. Sci. Comput.*, **26**(2):395–410, 2004.
69. He, W., Zeng, Y., and Li, G., An Adaptive Polynomial Chaos Expansion for High-Dimensional Reliability Analysis, *Struct. Multidiscip. Optimiz.*, **62**(4):2051–2067, 2020.
70. Diaz, P., Doostan, A., and Hampton, J., Sparse Polynomial Chaos Expansions via Compressed Sensing and D-Optimal Design, *Comput. Methods Appl. Mech. Eng.*, **336**:640–666, 2018.

71. Knio, O.M. and Le Maitre, O., Uncertainty Propagation in CFD Using Polynomial Chaos Decomposition, *Fluid Dyn. Res.*, **38**(9):616, 2006.
72. Constantine, P.G., Eldred, M.S., and Phipps, E.T., Sparse Pseudospectral Approximation Method, *Computer Methods in Applied Mechanics and Engineering*, **229**:1–12, 2012.
73. Conrad, P.R. and Marzouk, Y.M., Adaptive Smolyak Pseudospectral Approximations, *SIAM J. Sci. Comput.*, **35**(6):A2643–A2670, 2013.
74. Winokur, J., Kim, D., Bisetti, F., Le Maître, O.P., and Knio, O.M., Sparse Pseudo Spectral Projection Methods with Directional Adaptation for Uncertainty Quantification, *J. Sci. Comput.*, **68**(2):596–623, 2016.
75. Buzzard, G.T., Efficient Basis Change for Sparse-Grid Interpolating Polynomials with Application to T-Cell Sensitivity Analysis, *Comput. Biol. J.*, **2013**:562767, 2013.
76. Loukrezis, D. and De Gersem, H., Adaptive Sparse Polynomial Chaos Expansions via Leja Interpolation, *Math. Numer. Anal.*, arXiv:1911.08312, 2019.
77. Doostan, A. and Owhadi, H., A Non-Adapted Sparse Approximation of PDEs with Stochastic Inputs, *J. Comput. Phys.*, **230**(8):3015–3034, 2011.
78. Tsilifis, P., Huan, X., Safta, C., Sargsyan, K., Lacaze, G., Oefelein, J.C., Najm, H.N., and Ghanem, R.G., Compressive Sensing Adaptation for Polynomial Chaos Expansions, *J. Comput. Phys.*, **380**:29–47, 2019.
79. Rifkin, R.M. and Lippert, R.A., Notes on Regularized Least Squares, Tech. Rep., MIT-CSAIL-TR-2007-025, 2007.
80. Bottou, L., Large-Scale Machine Learning with Stochastic Gradient Descent, in *Proc. of COMPSTAT'2010*, Berlin: Springer, pp. 177–186, 2010.
81. Mao, X., Sabanis, S., and Renshaw, E., Asymptotic Behaviour of the Stochastic Lotka–Volterra Model, *J. Math. Anal. Appl.*, **287**(1):141–156, 2003.
82. Hundsdorfer, W. and Verwer, J.G., *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Vol. 33, Berlin: Springer Science & Business Media, 2013.
83. Alnæs, M., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., and Wells, G.N., The FEniCS Project Version 1.5, *Arch. Numer. Software*, **3**(100):20553, 2015.