Reliability Engineering and System Safety 252 (2024) 110459

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Probabilistic Performance-Pattern Decomposition (PPPD): Analysis framework and applications to stochastic mechanical systems

Ziqi Wang [a,*], Junho Song [b], Marco Broccardo [c,*]

[a] Department of Civil and Environmental Engineering, University of California, Berkeley, United States
[b] Department of Civil and Environmental Engineering, Seoul National University, South Korea
[c] Department of Civil, Environmental and Mechanical Engineering, University of Trento, Italy

ARTICLE INFO

Keywords:
Autoencoder
Clustering
Diffusion map
Manifold learning
Monte Carlo simulation
Pattern recognition
Stochastic dynamics
Uncertainty quantification

ABSTRACT

Numerous research efforts have been devoted to developing quantitative solutions to stochastic mechanical systems. In general, the problem is perceived as "solved" when a complete or partial probabilistic description on quantities of interest (QoIs) is determined. However, in the presence of complex system behavior, there is a critical need to go beyond computing probabilities. In fact, to gain a better understanding of the system, it is crucial to extract physical characterizations from the probabilistic structure of the QoIs, especially when the QoIs are computed in a data-driven fashion. Motivated by this perspective, the paper proposes a framework to obtain structuralized characterizations on behaviors of stochastic systems. The framework is named Probabilistic Performance-Pattern Decomposition (PPPD). PPPD analysis aims to decompose complex response behaviors, conditional to a prescribed performance state, into meaningful patterns in the space of system responses, and to investigate how the patterns are triggered in the space of basic random variables. To illustrate the application of PPPD, the paper studies three numerical examples: (1) an illustrative example with hypothetical stochastic processes input and output; (2) a stochastic Lorenz system with periodic as well as chaotic behaviors; and (3) a simplified shear-building model subjected to a stochastic ground motion excitation.

1. Introduction

The study of classical mechanics under uncertainty has become a significant area of research in engineering, attracting increasing attention and diverse applications. Over time, this field has expanded to encompass theoretical investigations into stochastic differential equations [1–7], stochastic dynamics [8–13], and risk and reliability theory [14–17], among others. Concurrently, the number of engineering applications in this domain has surged. Notable examples include [18–23].

Likewise in deterministic mechanical systems, the analysis of stochastic mechanical systems begins with the equation of motion, typically expressed as stochastic differential equations. However, unlike deterministic systems, a stochastic mechanical system is characterized by a set of random variables (which may be infinite) and produces a probabilistic description of Quantities of Interest (QoIs) as its output. Finding even approximate solutions can be challenging [1,2]. An alternative approach is to reformulate the equation of motion as an equation for the probability density function (such as the Smoluchowski or Fokker–Planck equation [24]), or as an equation for statistical moments (such as the moment closure method [25]). Unfortunately, these equations are also difficult to solve for complex multi-degree-of-freedom systems. In recent years, the development and formalization of Uncertainty Quantification (UQ) [26–30] have provided a new framework for addressing stochastic mechanical systems. UQ methods allow the problem to be approached as a forward UQ problem, with non-intrusive UQ methods [31,32] offering a practical solution. These methods treat the deterministic solution of the governing equations as a "black box" and focus on the statistical analysis of the input-QoI relationships. This is particularly useful in engineering applications with complex legacy codes that are challenging to adapt for intrusive UQ analysis. Given these advancements and the rise of high-performance computing [33–36], Monte Carlo simulation methods [37–44] and metamodeling [45–53] have become increasingly popular for forward UQ analysis and, consequently, for solving stochastic mechanical problems.

The use of classic non-intrusive UQ methods has been a remarkable advancement for the solution of stochastic mechanical problems. However, it has also stimulated an undesirable consequence: the problem of interest is perceived as "solved" once the probabilistic characterization of the QoIs is obtained. In fact, the classic UQ analysis and the following decision-making process are based on the statistics of the QoIs, losing *de facto* the physical information of the mechanical

* Corresponding authors.
E-mail addresses: ziqiwang@berkeley.edu (Z. Wang), junhosong@snu.ac.kr (J. Song), marco.broccardo@unitn.it (M. Broccardo).

https://doi.org/10.1016/j.ress.2024.110459
Received 26 November 2022; Received in revised form 4 August 2024; Accepted 20 August 2024
Available online 26 August 2024

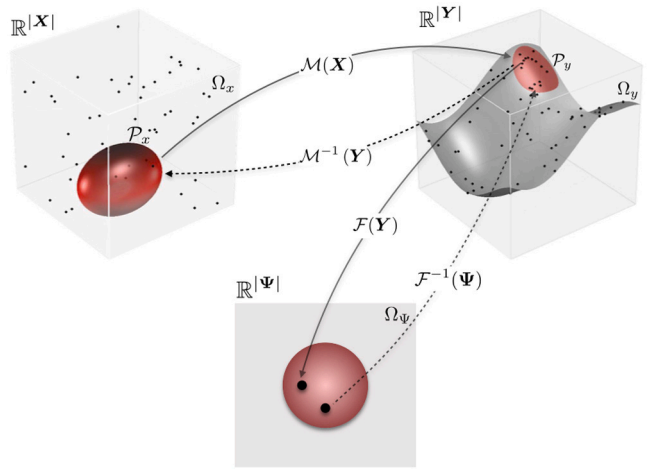

Fig. 1. Principles of PPPD. *Direct arrows denotes direct transformations, dashed lines denotes inverse reconstructions*.

problem (hidden within the black-box solver). There is a critical missing link in this context, that is extracting physical information and patterns from the probabilistic characterization of the QoIs. This is crucial, especially in the presence of multi-degree-of-freedom (MDoF) systems with complex behavior. A notable example in earthquake engineering is the stochastic dynamic analysis of MDoF systems with first-passage failure criterion, where the aim is to quantify the events that some response quantities exceed specified thresholds during a time interval. Computational methods for this problem focus on computing the first-passage probability; however, detailed analysis on the possible patterns and contributing factors of a first-passage event is rarely seen in the existing literature. Such analysis is important for engineers because it can facilitate the development of specialized design and retrofitting solutions against specific failure patterns. In contrast, an aggregated numeric output, like first-passage probability, does not provide this information.

This study aims to establish a formal framework for deriving a comprehensive physical characterization from the probabilistic structure of Quantities of Interest (QoIs). Within the scope of non-intrusive Uncertainty Quantification (UQ), the goal is to facilitate a physics-informed decision-making process. This approach emphasizes not only the statistical aspects of QoIs but, more crucially, the physical patterns that underlie their probabilistic representations. We introduce this framework as Probabilistic Performance-Pattern Decomposition (PPPD). The term "performance-pattern" is used because the framework focuses on behaviors defined by performance states. Specifically, the study develops methods to analyze the behavior patterns of complex stochastic systems and to identify key domains of basic random variables (which are the sources of randomness) that give rise to these patterns. The response of the system can be represented as a probabilistic reconstruction of the identified performance patterns.

The concept of establishing global characterizations of the behaviors of stochastic mechanical systems has been investigated in the literature. For example, in the study of stochastic differential equations, concepts such as random attractors and invariant manifolds have been developed as global characterizations of stochastic systems [54,55]. However, definitions and identifications of random attractors or invariant manifolds involve sophisticated (and often delicate) mathematical considerations, and applications of these concepts to real engineering problems are rare, and generally difficult to cast in a non-intrusive framework. Moreover, for stochastic systems without random attractors or invariant manifolds, they still may exhibit qualitatively different behaviors subjected to certain domains of random input, and thus there are needs to systematically analyze these behaviors. Another example for global characterization of stochastic mechanical systems is the concept of mutually exclusive and collectively exhaustive (MECE) set in system reliability theory [56,57]. In a system reliability approach to stochastic mechanical problems, the state space of a mechanical system is partitioned into various performance state levels (e.g., failure or safe in a 2-level partition), and the performance state of the system is contributed by combinations of performance states of components. The MECE set of a system performance state is a set of component performance states to give rise to the system performance state, and the set is MECE. Simply put, the MECE set of a system performance state corresponds to qualitatively different ways (with respect to component performance states) to achieve a system performance state. The limitation of the MECE set concept is that it is useful only if behaviors of a mechanical system can be meaningfully decomposed as combinations of behaviors of components, and such a decomposition should be a prior knowledge. By contrast, the concept of performance-patterns developed in this study is independent of decomposition of the system. In general, compared with random attractors, invariant manifolds, and MECE set, the concept of performance-pattern is more flexible and has promising potential as an effective analysis framework of a large variety of stochastic (not necessarily mechanical) systems.

In the context of structural reliability, there is another original contribution, which attempts to extract physical characterizations from the performance of a stochastic mechanical system. Starting from the idea of critical excitation [58], the Tail Equivalent Linearization Method (TELM) [59,60] was developed to study the reliability of hysteretic mechanical systems. The method, however, goes beyond the statistics of QoIs, as it provides a characterization of the nonlinear mechanical system in terms of a nonparametric Green's function, the critical excitation (named design point excitation), and the associated design point response. The method later has been proved to be successful in several applications (e.g., [61–64]). However, TELM is intrusive and is confined to a particular range of systems (e.g., softening and non-degradable systems, and first-order differentiable systems with respect to the input random variables). On the other hand, PPPD is free from these limitations, and generalizes the original idea of TELM for multiple patterns and generic mechanical systems.

A flourishing area of research closely related to PPPD is the implementation of manifold learning [65] in the context of UQ analysis, such as Probabilistic Learning on Manifolds (PLoM) [66–69] and surrogate modeling using manifold learning [70–75]. PLoM generates realizations from a latent space to mimic the unknown probability distribution of a dataset consisting of both the input and output of a computational model. Manifold learning-based surrogate modeling leverages hidden low-dimensional structures in the high-dimensional input and/or output of a computational model and builds surrogate models in the latent space. PPPD differs from these approaches as it does not focus on surrogate modeling but aims to uncover multiple physical patterns with associated uncertainties. It is worth mentioning that various manifold learning methods can be seamlessly integrated into the PPPD framework to enhance its capabilities for both forward and inverse uncertainty quantification in relation to the system's physical patterns.

Finally, it is essential to note that the goal of the proposed framework is fundamentally different from that of sensitivity analysis in uncertainty quantification (UQ). In typical sensitivity analysis, the objective is to determine which input random variable contributes the most to the variability of the quantities of interest (QoIs). While this technique is crucial for understanding system behavior, it remains incomplete from a physical perspective as it does not reveal the physical patterns underlying the probabilistic structure of the QoIs. Furthermore, it did not escape our attention that the proposed framework can be utilized in a fully data-driven manner. In this approach, large datasets (whether real or synthetic) of inputs and outputs are used to uncover patterns and regularities, facilitating the construction of data-driven models.

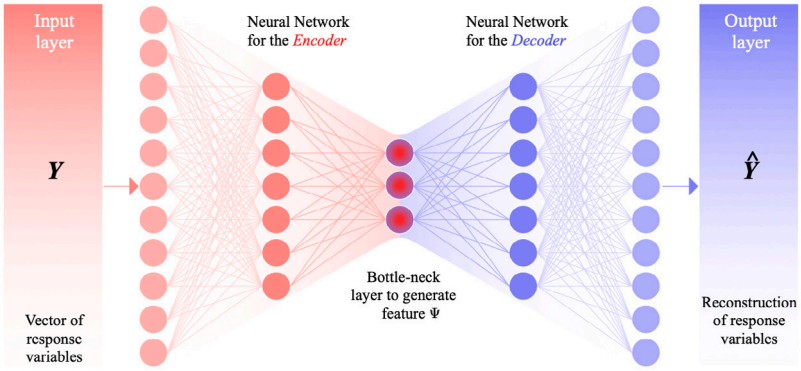

Fig. 2. Autoencoder in PPPD analysis. *The basic idea is: the output of the bottleneck layer should contain salient features of the original input, otherwise the reconstruction cannot be satisfactory*.

The structure of this paper is as follows: Section 2 introduces the general concepts underpinning the proposed framework. Sections 3 and 4 respectively develop the theoretical and computational frameworks of probabilistic performance-pattern decomposition (PPPD). Section 5 briefly discusses the nature and origins of performance patterns. Section 6 applies the developed methods to the analysis of various mechanical stochastic systems. Finally, Section 7 presents concluding remarks and directions for future research.

2. General principles of PPPD

Consider a mechanical system with a finite[1] set of basic random variables, denoted by $\boldsymbol{X}$. The basic random variables correspond to the source of randomness[2] for the system being considered. In general, the basic random variables involve epistemic uncertainty as well as aleatory variability present within the system or/and external excitation. For classical mechanical systems $\boldsymbol{X}$ may include variables of material properties, geometric quantities, initial and boundary conditions, dynamic excitation, and environmental effects. The complete description of stochastic dynamic systems is given by the joint probability distribution of the state variables (i.e., momentum and position) of all degree-of-freedom. However, in engineering applications the system behavior is usually (and better) described by a finite set of response variables, here denoted by $\boldsymbol{Y}$ (which are function of state variables). In this study, response variables are considered instead of state variables because response variables (by definition) are a direct description on the engineering behavior of interest. Since the source of randomness is captured by the basic random variables, the random response variables are deterministic function of basic random variables, i.e.,

$$\mathcal{M} : \boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} \mapsto \boldsymbol{y} \in \mathbb{R}^{|\boldsymbol{Y}|}, \tag{1}$$

where $\mathbb{R}^{|\boldsymbol{a}|}$ denotes the real number vector space with $|\boldsymbol{a}|$ being the dimension of vector $\boldsymbol{a}$. In general, Eq. (1) is a nonlinear mapping (not necessarily injective) from outcomes of $\boldsymbol{X}$ to outcomes of $\boldsymbol{Y}$, and the dimensions of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are generally different. Note that for time variant systems we consider the variable time to be included in definitions of $\boldsymbol{X}$ and $\boldsymbol{Y}$ (e.g., $\boldsymbol{X}$ and $\boldsymbol{Y}$ can represent discretized stochastic processes).

In engineering applications, it is meaningful to introduce performance states of the response variables. For example, in the design and safety assessment of structural systems, it is vital to know how they behave under different load and system properties. For these cases, a performance state can be introduced to focus on critical domains of the response variables. More abstractly, a performance state of the response variables $\boldsymbol{Y}$ is defined as a measurable event $\mathcal{P}_{\boldsymbol{Y}} \subseteq \mathbb{R}^{|\boldsymbol{Y}|}$. Then, the preimage of $\mathcal{P}_{\boldsymbol{Y}}$ in the sample space of $\boldsymbol{X}$ is $\mathcal{P}_{\boldsymbol{X}} := \left\{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} : \mathcal{M}(\boldsymbol{x}) \in \mathcal{P}_{\boldsymbol{Y}}\right\}$.

Provided with the joint probability density function (PDF) of the basic random variables $\boldsymbol{X}$, denoted by $f_{\boldsymbol{X}}(\boldsymbol{x})$, the joint PDF of $\boldsymbol{X}$ conditional on $\mathcal{P}_{\boldsymbol{X}}$ is

$$f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}}) = \frac{\mathbb{1}\left(\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{X}}\right) f_{\boldsymbol{X}}(\boldsymbol{x})}{\int_{\mathbb{R}^{|\boldsymbol{X}|}} \mathbb{1}\left(\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{X}}\right) f_{\boldsymbol{X}}(\boldsymbol{x})\, d\boldsymbol{x}}, \tag{2}$$

where $\mathbb{1}\left(\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{X}}\right)$ is a binary indicator function that gives "1" if $\boldsymbol{x} \in \mathcal{P}_{\boldsymbol{X}}$, and "0" the otherwise. Using Eq. (1), the joint PDF of response variables $\boldsymbol{Y}$ conditional on the performance state $\mathcal{P}_{\boldsymbol{Y}}$ can be expressed by the multiple integral

$$f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}}) = \int_{\mathbb{R}^{|\boldsymbol{X}|}} \delta(\boldsymbol{y} - \mathcal{M}(\boldsymbol{x})) f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})\, d\boldsymbol{x}, \tag{3}$$

where $\delta(\cdot)$ is the Dirac-Delta function.

If the joint PDF $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ can be obtained from Eq. (3), a complete statistical description on the response variables $\boldsymbol{Y}$ within a specified performance state is available. However, for nontrivial problems (e.g., problems with $\mathcal{M}$ being nonlinear and computationally demanding, and/or dimensionality of $\boldsymbol{X}$ or $\boldsymbol{Y}$ being high), a direct evaluation of Eq. (3) is infeasible. As a consequence, instead of attempting to obtain $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$, a common practice is to study statistical properties of $\boldsymbol{Y}|\mathcal{P}_{\boldsymbol{Y}}$ using mean, covariance matrix, marginal distributions, and other statistical measures that are relatively convenient to obtain.

In this study, an alternative path to systematically investigate the probabilistic structure of the response variables $\boldsymbol{Y}$ (within a performance state) is explored. Moreover, this study goes beyond a statistical characterization of the response variables, since the critical domain of the random input (i.e., basic random variables $\boldsymbol{X}$) that generates the probabilistic structure of $\boldsymbol{Y}$ will also be investigated. Specifically, given the performance state of interest, this paper studies the procedure of (a) determining meaningful patterns for response variables, and (b) determining the critical domain of basic random variables that triggers each pattern. This procedure is defined as Probabilistic Performance Pattern Decomposition (PPPD).

Fig. 1 illustrates the principles of PPPD analysis. It involves the interplay between the basic variables space and the response variables space. Moreover, a feature space is introduced to uncover performance patterns. The basic ingredients of PPPD are described as follows: (a) The basic random variables $\boldsymbol{X}$ are mapped to the response variables $\boldsymbol{Y}$ through a model $\mathcal{M}$. Since a performance state is of interest, PPPD focuses on $\mathcal{P}_{\boldsymbol{X}}$ and $\mathcal{P}_{\boldsymbol{Y}}$ (marked as red regions in the figure). (b) The structure of $\mathcal{M}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$ is typically well-hidden in its embedded (possibly

[1] If a system with an infinite number of random variables is of interest (e.g., systems involve random processes/fields), for practical purposes one could discretize the random processes/fields by a finite set of random variables.

[2] The definition for "source of randomness" is subjected to confinement on the specific physical/mathematical models used to describe the problem.

high-dimensional) $\boldsymbol{Y}$ space. To disclose its structure, we perform manifold learning via constructing a nonlinear feature projection $\mathcal{F}: \boldsymbol{Y} \mapsto \boldsymbol{\psi}$ to identify meaningful patterns. The patterns identified in the feature space are then mapped back to the $\boldsymbol{Y}$ space. (c) With the knowledge of the performance patterns in the space of $\boldsymbol{Y}$, we finally identify the critical regions which trigger each pattern in the space of $\boldsymbol{X}$.

3. Mathematical formulations of PPPD

3.1. Probabilistic decomposition

We investigate the structure of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ by introducing a latent random variable[3] Z defined in an auxiliary sample space Ω_Z with distribution function $F_Z(z)$. We define $F_Z(z)$ as the latent distribution and construct a joint distribution between the vector $\boldsymbol{Y} \in \mathcal{P}_{\boldsymbol{Y}}$ and the latent variable Z defined in the augmented sample space $\Omega_{\mathcal{P}_{\boldsymbol{Y}}} \times \Omega_Z$ (where we introduce $\Omega_{\mathcal{P}_{\boldsymbol{Y}}}$ to denote the sample space of the performance state $\mathcal{P}_{\boldsymbol{Y}}$). It follows that $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ can be written as[4]:

$$f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}}) = \mathbb{E}_Z\left[f_{\boldsymbol{Y}}(\boldsymbol{y}|Z,\mathcal{P}_{\boldsymbol{Y}})\right] = \int_{\Omega_Z} f_{\boldsymbol{Y}}(\boldsymbol{y}|Z,\mathcal{P}_{\boldsymbol{Y}})\, dF_Z(z), \tag{4}$$

where $\mathbb{E}_Z$ denotes expectation with respect to the latent variable. Observe that Eq. (4) can be interpreted as the Fredholm integral equation of the first kind, where $f_{\boldsymbol{Y}}(\boldsymbol{y}|Z,\mathcal{P}_{\boldsymbol{Y}})$ is the kernel function. Now consider a partition of Ω_Z into a finite set of K mutually exclusive and collective exhaustive events, i.e., $\Omega_Z = \cup_{k=1}^{K} E_k$, $E_k \cap E_l = \emptyset$, $k \neq l$, and $K \in \mathbb{N}^+$, and define $\lambda_k \equiv \mathbb{P}(Z \in E_k)$ and $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}}) \equiv f_{\boldsymbol{Y}}(\boldsymbol{y}|E_k,\mathcal{P}_{\boldsymbol{Y}})$. Given this, Eq. (4) can be rewritten to:

$$f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}}) = \mathbb{E}_Z\left[f_{\boldsymbol{Y}}(\boldsymbol{y}|Z,\mathcal{P}_{\boldsymbol{Y}})\right] = \sum_{k=1}^{K} \lambda_k f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}}). \tag{5}$$

The density $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ is defined as the kth component density and $\lambda_k > 0$, $\sum_{k=1}^{K} \lambda_k = 1$, is defined as the kth component weight. The component density $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ is the likelihood of the realization of $\boldsymbol{Y}$ conditional to the event E_k and performance state $\mathcal{P}_{\boldsymbol{Y}}$, while the component weight λ_k provides a direct measure on the importance of $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$. Note that K is generally unknown and to be determined in the PPPD procedure. Although Eqs. (4) and (5) are equivalent, Eq. (5) offers the advantage of highlighting the decomposition of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ into a finite set of component densities.

It is important to note that different from mixture model approximation to distribution functions, Eq. (5) is by construction exact. Although the structure of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ is an unknown to be disclosed, we assume the structure exists in the space of latent variable. Provided Eq. (5) to be a formalization for the concept of performance pattern, and given $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ to be decomposed in a conceptually meaningful way,[5] we define the component density $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ to be the PDF of the kth performance pattern, the component weight λ_k to be the relative importance of the kth performance pattern, and event E_k^Z to be the label of the kth performance pattern. Moreover, the mean of $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ can be regarded as a characteristic vector to represent the performance pattern. Note that although Eq. (5) provides no hints on how to decompose $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ for a specific application, we are interested in the nontrivial cases for which $K > 1$ and $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}}) \neq f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$.

For a given realization of $\boldsymbol{y}^{(i)}$, $\boldsymbol{y}^{(i)}$ belongs to the kth performance pattern with the likelihood:

$$\mathcal{L}(k) = \mathbb{P}(k|\boldsymbol{y}^{(i)},\mathcal{P}_{\boldsymbol{Y}}) = \frac{\lambda_k f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|k,\mathcal{P}_{\boldsymbol{Y}})}{f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|\mathcal{P}_{\boldsymbol{Y}})} = \frac{\lambda_k f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|k,\mathcal{P}_{\boldsymbol{Y}})}{\sum_{j=1}^{K} \lambda_j f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|j,\mathcal{P}_{\boldsymbol{Y}})}. \tag{6}$$

The likelihood $\mathcal{L}(k)$ can be zero if $\boldsymbol{y}^{(i)} \notin \Omega_{\mathcal{P}_{\boldsymbol{Y}},k}$, where $\Omega_{\mathcal{P}_{\boldsymbol{Y}},k}$ denotes the sample space of $f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|k,\mathcal{P}_{\boldsymbol{Y}})$. A zero likelihood also implies $f_{\boldsymbol{Y}}(\boldsymbol{y}^{(i)}|k,\mathcal{P}_{\boldsymbol{Y}})$ is a truncated distribution, i.e. $\Omega_{\mathcal{P}_{\boldsymbol{Y}},k} \subset \Omega_{\mathcal{P}_{\boldsymbol{Y}}}$. If the component densities are truncated distributions, the performance patterns provide a "hard decomposition" (partition) of $\mathcal{P}_{\boldsymbol{Y}}$, otherwise they provide a "soft decomposition" in which each realization of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ has a nonzero probability to belong to any of the patterns.

Now, to identify the critical domains of basic random variables that trigger each performance pattern, we can use Bayes' rule as follow

$$f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}}) = \frac{\mathbb{P}(k|\boldsymbol{x},\mathcal{P}_{\boldsymbol{X}}) f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})}{\int_{\mathbb{R}^{|\boldsymbol{X}|}} \mathbb{P}(k|\boldsymbol{x}',\mathcal{P}_{\boldsymbol{X}}) f_{\boldsymbol{X}}(\boldsymbol{x}'|\mathcal{P}_{\boldsymbol{X}}) d\boldsymbol{x}'}. \tag{7}$$

Observe that for a given sample $\boldsymbol{x}^{(i)}$, $\mathbb{P}(k|\boldsymbol{x}^{(i)},\mathcal{P}_{\boldsymbol{X}}) \equiv \mathbb{P}(k|\boldsymbol{y}^{(i)} = \mathcal{M}(\boldsymbol{x}^{(i)}),\mathcal{P}_{\boldsymbol{Y}} = \mathcal{M}(\mathcal{P}_{\boldsymbol{X}})) = \mathcal{L}(k)$. Therefore, we can write:

$$f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}}) = \frac{1}{\mathcal{K}} \frac{f_{\boldsymbol{Y}}(\mathcal{M}(\boldsymbol{x})|k,\mathcal{P}_{\boldsymbol{Y}})}{f_{\boldsymbol{Y}}(\mathcal{M}(\boldsymbol{x})|\mathcal{P}_{\boldsymbol{Y}})} f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}}), \tag{8}$$

where $\mathcal{K}$ is a normalizing constant. $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$ is named the generating density for the kth performance pattern. Similar to Eq. (5), $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$ can be written in the decomposition form:

$$f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}}) = \sum_{k=1}^{K} \lambda_k f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}}). \tag{9}$$

Note that the λ_k in Eqs. (5) and (9) are, by definition, identical.

3.2. Feature space representation

Eq. (5) should be constructed such that the performance patterns are "manifestly different" from each other. To define manifestly different, we first introduce the feature mapping of $\boldsymbol{Y}$ described as:

$$\begin{aligned} &\boldsymbol{F}: \boldsymbol{y} \in \mathbb{R}^{|\boldsymbol{Y}|} \mapsto \boldsymbol{\psi} \in \mathbb{R}^{|\boldsymbol{\Psi}|} \\ &\boldsymbol{F}^{-1}: \boldsymbol{\psi} \in \mathbb{R}^{|\boldsymbol{\Psi}|} \mapsto \hat{\boldsymbol{y}} \in \mathbb{R}^{|\boldsymbol{Y}|} \end{aligned} \tag{10}$$

where the dimension of the feature vector $\boldsymbol{\psi}$ is typical much smaller than $\boldsymbol{y}$. Note that $\boldsymbol{F}^{-1}$ represents the (approximate) reconstruction function rather than the inverse function, because the inverse function may not exist. Ideally, we seek a low-dimensional feature vector preserving salient properties of the original vector. In practice, the trade-off between the dimension of $\boldsymbol{\psi}$ and the loss of information is typically balanced by setting a threshold for the reconstruction error of the dimensionality reduction algorithm. The feature mapping is introduced to disclose the structure of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$, and thus in the feature space, similar to Eq. (5), the decomposition is:

$$f_{\boldsymbol{\Psi}}(\boldsymbol{\psi}|\mathcal{P}_{\boldsymbol{\Psi}}) = \sum_{k=1}^{K} \lambda_k f_{\boldsymbol{\Psi}}(\boldsymbol{\psi}|k,\mathcal{P}_{\boldsymbol{\Psi}}), \tag{11}$$

where the λ_k is, by construction, identical to that in Eqs. (5) and (9). This is because the weight of a performance pattern is a property of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$, which should be preserved by construction in the feature space designed to uncover the pattern and in the preimage of $\mathcal{P}_{\boldsymbol{Y}}$ that generates the pattern. The corresponding mathematical construction is: the same latent Z as well as its partition $\{E_1, E_2, \ldots, E_K\}$ is used to augment $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{\Psi}$, so that the weight $\lambda_k \equiv \mathbb{P}(Z \in E_k)$ is maintained. A natural requirement for the projected performance patterns $f_{\boldsymbol{\Psi}}(\boldsymbol{\psi}|k,\mathcal{P}_{\boldsymbol{\Psi}})$ is: the expected within-pattern distance should be smaller than the expected between-pattern distance. This is a widely adopted principle in clustering analysis.

[3] In physics, latent variables are sometimes introduced to reflect the tangible effects of hidden mechanisms which are difficult to observe (but in principle can be observed). In this paper, however, the latent variable is introduced to reflect the abstract concept of functional structure of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$.

[4] This equation posits $Q(Z) = Q(Z|\mathcal{P}_{\boldsymbol{Y}})$, because the sole purpose of introducing Z is to identify patterns in $\mathcal{P}_{\boldsymbol{Y}}$.

[5] By "conceptual meaningful", we indicate that the resulting performance patterns are manifestly different from each other for a specific application.

4. Computational framework of PPPD

4.1. Realizations of basic and response random variables

In this section, we introduce the computational framework for PPPD based on sampling methods. Specifically, the framework is developed using a dataset of random realizations of $\boldsymbol{X}$ drawn from $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$ and the corresponding realizations of $\boldsymbol{Y} = \mathcal{M}(\boldsymbol{X})$. For later use, we introduce the limit-state surface to describe the boundary of the performance state $\mathcal{P}_{\boldsymbol{X}}$:

$$G(\boldsymbol{y}) = 0\,, \tag{12}$$

and $G(\boldsymbol{y}) \le 0$ denotes that $\boldsymbol{y}$ is within $\mathcal{P}_{\boldsymbol{Y}}$; therefore,

$$\mathcal{P}_{\boldsymbol{Y}} = \left\{\boldsymbol{y} \in \mathbb{R}^{|\boldsymbol{Y}|} : G(\boldsymbol{y}) \le 0\right\}\,. \tag{13}$$

The preimage $\mathcal{P}_{\boldsymbol{X}}$ can be expressed as:

$$\mathcal{P}_{\boldsymbol{X}} = \left\{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} : G(\mathcal{M}(\boldsymbol{x})) \le 0\right\}\,. \tag{14}$$

A naïve rejection sampling can generate samples from $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$, such that it samples from $f_{\boldsymbol{X}}(\boldsymbol{x})$ and only saves the points with $G(\mathcal{M}(\boldsymbol{x})) \le 0$. This approach is effective if $\mathbb{P}(\boldsymbol{X} \in \mathcal{P}_{\boldsymbol{X}})$ is relatively large. If $\{\boldsymbol{X} \in \mathcal{P}_{\boldsymbol{X}}\}$ is a rare event, the rejection sampling becomes infeasible. For rare event simulations, an attractive approach is the sequential Monte Carlo (SMC)/Subset Simulation [40,41,76–79]. A key concept in SMC is to introduce a finite sequence of intermediate states, denoted by $\mathcal{P}_{\boldsymbol{X}}^{(j)}$, $j = 1, 2, \ldots, m$, that satisfies:

$$\mathcal{P}_{\boldsymbol{X}}^{(1)} \supset \mathcal{P}_{\boldsymbol{X}}^{(2)} \supset \cdots \supset \mathcal{P}_{\boldsymbol{X}}^{(m)} = \mathcal{P}_{\boldsymbol{X}}\,. \tag{15}$$

The intermediate states can be constructed by:

$$\mathcal{P}_{\boldsymbol{X}}^{(j)} = \left\{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} : G(\mathcal{M}(\boldsymbol{x})) \le g^{(j)}\right\}\,, \tag{16}$$

where $g^{(j)}$ is a relaxation parameter that is monotonic with j, i.e., $g^{(1)} > g^{(2)} > \cdots > g^{(m)} = 0$. The constructions of Eqs. (15) and (16) are not unique, alternative ways to relax the target sampling domain are studied in [41]. The relaxation parameter can be either pre-specified using certain rule of thumbs [78] or selected adaptively such that the probability $\mathbb{P}(\boldsymbol{X} \in \mathcal{P}_{\boldsymbol{X}}^{(j+1)} | \boldsymbol{X} \in \mathcal{P}_{\boldsymbol{X}}^{(j)})$ is relatively large [76,77]. In each intermediate step of the SMC simulation, a Markov Chain Monte Carlo (MCMC) sampling is performed starting with seed samples within $\mathcal{P}_{\boldsymbol{X}}^{(j)}$ to generate samples for $\mathcal{P}_{\boldsymbol{X}}^{(j+1)}$. An SMC procedure that adaptively specify $\mathcal{P}_{\boldsymbol{X}}^{(j)}$ to sample from $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$ is described in Appendix A.

The general principle for selecting the sample size is to generate a sufficient number of representative points to ensure that the identified dominant patterns are stable. This principle shares a common flaw with global optimization: for a generic continuous problem, it is nearly impossible to enumerate all patterns/optima. However, unlike optima in global optimization, performance patterns have distinct statistical meanings. For example, if 10^3 representative samples are generated from a performance state, the patterns that are missed will have low probabilities/importance compared to the identified patterns, proportional to 10^{-3}. In the numerical examples studied in this paper, the typical sample size is 2000.

4.2. Feature mapping via manifold learning

Let $\mathcal{Y} = \{\boldsymbol{y}^{(i)}\}_{i=1}^{N}$ denote the dataset of N samples from $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$. We apply feature mapping Eq. (10) to cast them into a low-dimensional feature space. Note that if $\boldsymbol{Y}$ involves components from different sources with different scales, it can be beneficial to perform normalization on $\mathcal{Y}$ before the feature space transformation. In this paper, two nonlinear feature mappings based on manifold learning will be investigated, but there exists alternative options [49,80,81].

4.2.1. Diffusion map

Diffusion maps [82,83] are a manifold learning method for reducing the dimensionality of data while preserving its essential structure. The dimensionality reduction is based on the eigenfunctions of a Markov matrix, which describes affinities within a dataset. As a manifold learning technique, diffusion maps can uncover the underlying manifold from which the data has been sampled. In the context of physics-driven problems, diffusion maps can reveal dominant physical patterns. Central to this technique is the concept of diffusion distance, which measures the similarity between data points based on their connectivity and relationships. In the reduced-dimensional space created by diffusion maps, known as the embedding space, this diffusion distance is defined as the Euclidean distance, a noise-insensitive metric [82]. This means that in the embedding space, the measure of similarity between samples, which aims at capturing the intrinsic geometry of the data, is defined by the simple L_2 norm of the vector connecting the two points. In the context of physics-based problems, the similarity uncovered by diffusion maps corresponds to the underlying physical patterns, providing insights into the fundamental behaviors and properties of the system under study. Detailed implementation steps for applying diffusion maps to PPPD can be found in Appendix B.

4.2.2. Autoencoder

The autoencoder [84,85] is a manifold leaning method that uses feed-forward neural network to generate simplified encoding of a dataset. In the context of PPPD analysis, an autoencoder consists of an encoder which maps each response vector $\boldsymbol{y}^{(i)}$ into a feature vector $\boldsymbol{\psi}^{(i)}$, and a decoder which maps $\boldsymbol{\psi}^{(i)}$ back to a reconstruction of $\boldsymbol{y}^{(i)}$, denoted as $\hat{\boldsymbol{y}}^{(i)}$. The autoencoder is trained to minimize the distance between $\boldsymbol{y}^{(i)}$ and $\hat{\boldsymbol{y}}^{(i)}$, i.e., the reconstruction error. The basic concept of the autoencoder in PPPD analysis is illustrated in Fig. 2. Owing to the flexibility of artificial neural network techniques, compared with the diffusion map, autoencoder can be more powerful in processing complex dataset, at the cost of training effort. Implementation details of the autoencoder for PPPD can be found in Appendix C.

4.3. Performance pattern identification

Given the set of feature vectors $\boldsymbol{\Psi}$, the subsequent step of PPPD is to find patterns in $\boldsymbol{\Psi}$. The aim is to restore the distribution $(Z, \boldsymbol{Y})$ from samples of $\boldsymbol{Y}$. This is a well-known unsupervised statistical learning problem that can be tackled by clustering analysis.

4.3.1. Determine the number of patterns & clustering analysis

If a manifold learning technique can effectively map the original samples into a two- or three-dimensional feature space, the number of patterns is expected to be visually identified. For a relatively high-dimensional feature space embedding, to determine the number of patterns, one can study a strict partitioning clustering problem described as follows.

Given a dataset $\{\boldsymbol{\psi}^{(i)}\}_{i=1}^{N}$, find a partition, denoted by $P = \left\{P_1, \ldots, P_{K^*}\right\}$, of the N samples into K^*, $K^* \le N$, subsets so as to minimize a specified measure of the partition:

$$P^* = \arg\min_{P} q(P)\,, \tag{17}$$

where the measure $q(\cdot)$ is defined to be independent of K^* so that K^* is also an unknown to be determined from Eq. (17). In clustering analysis practice, a two-step approach is typically used to solve Eq. (17). In the first step, a measure $q_K(P)$ is defined to find the optimal partition for a specified number of clusters. For example, in the classical k-means clustering, $q_K(P)$ is defined by the within-cluster sum of squares, i.e.,

$$q_K(P) = \sum_{i=1}^{K} \sum_{\boldsymbol{\psi}^{(j)} \in P_i} \left\|\boldsymbol{\psi}^{(j)} - \boldsymbol{\mu}_i\right\|^2\,, \tag{18}$$

where $\boldsymbol{\mu}_i$ is the mean of $\boldsymbol{\psi}^{(j)}$ in P_i. With $q_K(P)$ specified, the optimal partition for a specified K, denoted as P_K^*, is obtained from:

$$P_K^* = \arg\min_P q_K(P)\,. \tag{19}$$

Even though the optimization problem defined by Eq. (19) is usually NP-hard, various clustering algorithms [86] have proven to be effective for practical applications.

In the second step, a measure $\ell(P_K^*)$ is defined to find the optimal number of clusters, K^*, and consequently, the optimal partition P^* via:

$$\begin{aligned} K^* &= \arg\min_{K\in\mathbb{N}^+} \left\{\ell(P_K^*)\right\} \\ P^* &= P_{K=K^*}^* \end{aligned} \tag{20}$$

The specification of $\ell(\cdot)$ belongs to the problem of determining the "exact" number of groups in a dataset, which is a fundamental, yet largely unsolved challenge in clustering analysis. Numerous approaches to this problem have been suggested over the past decades [87–89]. Essentially, the "exact" number of groups is an ill-defined concept, unless an unambiguous criterion for similarity is proposed for a specific application. The approach considered in this paper is based on information theory [88], where $\ell(\cdot)$ is defined as:

$$\ell(P_K^*) = d_{K-1}^{-a} - d_K^{-a}\,, \tag{21}$$

where the transformation power a is typically set to $a = n/2$, in which n is the dimension, d_0 is defined to be 0, and d_K is the approximate distortion expressed by:

$$d_K = \frac{1}{n\cdot N}\min_{k=1,\ldots,K}\sum_{i=1}^{N}(\boldsymbol{\psi}^{(i)}-\boldsymbol{\mu}_k^{(i)})^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{\psi}^{(i)}-\boldsymbol{\mu}_k^{(i)})\,, \tag{22}$$

where $\boldsymbol{\Sigma}_k$ denotes the covariance matrix for the kth cluster and $\boldsymbol{\mu}_k^{(i)}$ denotes the cluster center that is closest to sample $\boldsymbol{\psi}^{(i)}$, for a specified k. The essential idea of the information theoretic approach is to use the K versus $\ell(P_K^*)$ curve to investigate the influence of number of clusters on the clustering quality. The distortion d_K is a measure of the within-cluster dispersion, and it is monotonically decreasing as K increases. The information theoretic approach assumes that if K is approaching the "true" number of clusters K^*, the drop in distortion will attain the maximum (and $\ell(P_K^*)$ will achieve the minimum), since adding more clusters beyond K^* simply partitions within rather than between groups.

Once the partition P^* is obtained, a participation factor of each cluster can be evaluated by:

$$\Gamma_k = \frac{\sum_{j=1}^{N}\mathbb{1}\left(\boldsymbol{\psi}^{(j)}\in P_k\right)}{N}\,. \tag{23}$$

The participation factor can be used as an approximation to the component weight λ_k of each performance pattern. Moreover, the mean vector of each cluster, or the sample closest to the mean vector, can be used as a characteristic vector to represent each performance pattern.

Other than hard clustering approaches, one could also use soft clustering algorithms [86] to establish a soft decomposition. Recalling concepts introduced in Section 3, a hard clustering corresponds to a hard decomposition (partition) of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$, and each sample can only belong to one of the patterns; while a soft clustering corresponds to a soft decomposition of $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$, and each sample is allowed to belong to more than one pattern.

4.3.2. Parametric description of performance patterns

Given the results of a clustering analysis, one could construct a parametric model to describe the component densities $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ for each cluster/performance pattern. A typical approach to construct a parametric PDF model is to use mixture distribution. Specifically, $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ can be written in terms of a parametric mixture model, and $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ is described by component of the parametric mixture model, i.e.

$$\begin{aligned} f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}}) &\cong \sum_{k=1}^{K^*}\hat{\lambda}_k f_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\theta}_k,\mathcal{P}_{\boldsymbol{Y}}) \\ f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}}) &\cong f_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\theta}_k,\mathcal{P}_{\boldsymbol{Y}}) \end{aligned} \tag{24}$$

where $\boldsymbol{\theta}$ is a set of global parameters, $\boldsymbol{\theta}_k$ is a set of component parameters, and $\hat{\lambda}_k$ is the component weight of the mixture model. Parameters of the mixture model can be estimated using the Expectation–Maximization (EM) algorithm [90], guided by the partition and labeled samples obtained from clustering analysis. Notice that, in principle, the mixture model can also be used directly for clustering. However, in the PPPD framework, we perform pattern recognition (e.g., with diffusion maps or autoencoders) in the original space. This space is typically high-dimensional for dynamical problems, making it intractable to use the EM algorithm directly. The density can also be constructed using a mixture model in the feature space. However, due to the unknown number of clusters, building a parametric model becomes more challenging and results in multiple solutions that are less interpretable. Once the number of patterns is identified, the probability decomposition in the input space is conveniently modeled by a density mixture, with parameter identification provided by the EM algorithm. In this case, the solution is unique as the number of clusters is known.

Given the component density $f_{\boldsymbol{Y}}(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\theta}_k,\mathcal{P}_{\boldsymbol{Y}})$, the generating density $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$ can be obtained by Eq. (8), in principle. However, since the model $\mathcal{M}(\cdot)$ is generally not explicit, Eq. (8) is particularly useful only when a Monte Carlo approach is employed to sample from $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$. If a parametric description of $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$ is of interest, one could employ the mixture model approach.

Finally, it is important to note that parametric descriptions of $f_{\boldsymbol{Y}}(\boldsymbol{y}|k,\mathcal{P}_{\boldsymbol{Y}})$ or $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$ are not always feasible. For generic problems incapable of parametrization, the numerical solutions obtained from clustering analysis can be regarded as the final output of PPPD analysis. In clustering analysis, instead of a parametric description, one could only obtain statistical/geometrical descriptions on each performance pattern and its generating density.

4.4. Procedures of PPPD

The basic computational procedures of PPPD analysis is summarized as follows.

Algorithm 1 Procedures of PPPD analysis

Step 1: Problem statement

- Define basic random variables $\boldsymbol{X}$, and define the joint PDF of $\boldsymbol{X}$.
- Define the response random variables $\boldsymbol{Y}$ to describe the behavior of the system.
- Specify the computational model $\mathcal{M}(\cdot)$ that maps $\boldsymbol{X}$ to $\boldsymbol{Y}$.
- Define the performance state of interest.

Step 2: Obtain random realizations of basic and response variables

- Draw N pair of samples from PDFs $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$ and $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$.

Step 3: Feature mapping

- Perform feature mapping on samples of $\boldsymbol{Y}$.

Step 4: Performance pattern identification

- Determine the number of performance patterns in the feature space.
- Extract the performance patterns of $\boldsymbol{Y}$ in the feature space and their generating densities $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$ via clustering analysis.
- (Optional) Obtain a parametric description on performance patterns of $\boldsymbol{Y}$ and their generating densities $f_{\boldsymbol{X}}(\boldsymbol{x}|k,\mathcal{P}_{\boldsymbol{X}})$.

5. Origin of performance patterns

In this Section, we discuss the possible causes that generate multiple performance patterns. Clearly, a necessary but not sufficient condition for observing multiple performance patterns is the random variability within the system or/and the external excitation, otherwise the performance of the system will be a deterministic event. Given that there is randomness involved, the origin of multiple performance patterns can be traced back to the following causes.

(a) *Source*: The existence of multiple patterns in the basic random variables.
(b) *Propagation*: The existence of bifurcations or discontinuities within the deterministic physical model.
(c) *Constraint*: The specific property of the performance state.
(d) *Subjectivity*: The specific property of the distance metric defined in feature mapping.

To understand "(c) *Constraint*", note that the performance state $\mathcal{P}_{\boldsymbol{Y}}$ applies a truncation to the original sample space of response variables, and after the truncation the conditional distribution $f_{\boldsymbol{Y}}(\boldsymbol{y}|\mathcal{P}_{\boldsymbol{Y}})$ could exhibit multiple patterns even if $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is unimodal. To understand "(d) *Subjectivity*", note that a redefinition of the distance metric alters the structure of the dataset, so that patterns that are not inherent in the original dataset could be triggered. In the feature mapping procedure, if a conventional distance metric is used (e.g., the Euclidean distance), the manifold learning technique could, at best, make patterns that are ambiguous in the original space easier to be identified. However, if a problem specific distance metric is used, the metric introduces additional prior knowledge (subjectivity) so that new patterns (that do not exist within the original dataset) could be triggered. It follows that the use of an inappropriate problem specific distance metric could produce artificial performance patterns, thus the use of problem specific distance metrics should be handled with cautiousness. In practical applications, various distance metrics can be used to examine the sensitivity of patterns to metric selection. If patterns are insensitive to the metric, the results are considered robust. However, if high sensitivity is detected, it becomes crucial to identify potential spurious modes and carefully consider the choice of metric. Moreover, using meaningful physics-informed distance metrics may assist in the discovery of important, well-hidden structures. Investigations on the use of physics-informed distance metrics will be addressed in follow-up studies. It can be concluded from this section that the performance pattern reflects not only the characteristics of the randomness source and the deterministic physical model but also encompasses properties of the parametric region of interest and understanding of system behaviors.

6. Numerical investigations

6.1. An illustrative example of simple system identification

To illustrate the main ideas and procedures of PPPD analysis, consider a hypothetical system with basic random variables $\boldsymbol{X}$ of the form $\boldsymbol{X} = [\boldsymbol{X}_p, X_{st}]$, where $\boldsymbol{X}_p = [X(t_1),\dots,X(t_n)]$, $X(t_i) \sim \mathcal{N}(0,1)$, $i = 1,\dots,n$, represents a discretized zero-mean Gaussian white noise, and $X_{st} \in \{1,2,3,4\}$, is a discrete uniform random variable. For a realization of X_{st}, the response of the hypothetical system, $\boldsymbol{Y} = [Y(t_1),\dots,Y(t_n)]$, is a discretized stochastic process expressed by:

$$\boldsymbol{Y} = \begin{cases} \sin(\pi \boldsymbol{t}/4) + \cos(\pi \boldsymbol{t}/3) + 0.3\boldsymbol{X}_p\,, & X_{st} = 1 \\ \sin(\pi \boldsymbol{t}/4 + 0.1) + \cos(\pi \boldsymbol{t}/3 - 0.1) + 0.3\boldsymbol{X}_p\,, & X_{st} = 2 \\ \sin(\pi \boldsymbol{t}/3) + \cos(2\pi \boldsymbol{t}/5) + 0.3\boldsymbol{X}_p\,, & X_{st} = 3 \\ \sin(\pi \boldsymbol{t}/3 + 0.1) + \cos(2\pi \boldsymbol{t}/5 - 0.1) + 0.3\boldsymbol{X}_p\,, & X_{st} = 4 \end{cases} \tag{25}$$

where $\boldsymbol{t} = [t_1,\dots,t_n]$. It is assumed the whole sample space of $\boldsymbol{Y}$ is of interest, i.e., $\mathcal{P}_{\boldsymbol{Y}} = \mathbb{R}^{|Y|}$. The time sequence $\boldsymbol{t}$ is set to starting from 0.01 s to 10 s, with a uniform incremental time step of 0.01 s. Therefore, the dimension of $\boldsymbol{Y}$ is 1000.

Assumed that one can only observe the input $\boldsymbol{X}$ and output $\boldsymbol{Y}$, without a prior knowledge on Eq. (25). The PPPD analysis is used to retrieve structuralized information from the dataset of $\boldsymbol{X}$ and $\boldsymbol{Y}$. To start the PPPD analysis, using a direct Monte Carlo simulation, 2000 random realizations of $\boldsymbol{Y}$ are obtained (shown in Fig. 3). It appears impossible to identify any pattern through visual inspection of Fig. 3.

Fig. 4 shows the 2000 realizations of $\boldsymbol{Y}$ embedded into a 3-dimensional feature space, obtained from diffusion maps with various time-scales τ. The similarity matrix is constructed using Eq. (B.1) with $\boldsymbol{L}^2$-norm distance and ϵ is set to 10. It can be observed from Fig. 4 that: (a) for a relatively high resolution embedding (a relatively small time-scale τ), four patterns can be identified in the feature space; (b) for a relatively low resolution embedding (a relatively large time-scale τ), two patterns can be identified. This observation implies: (a) there exists four patterns; (b) the four patterns can be divided into two groups, and for each group the similarity within the group is more significant than the similarity between groups.

Next, an autoencoder with 5 hidden layers and 100-30-3 neurons for each hidden layer of the encoder (the decoder is symmetric) is

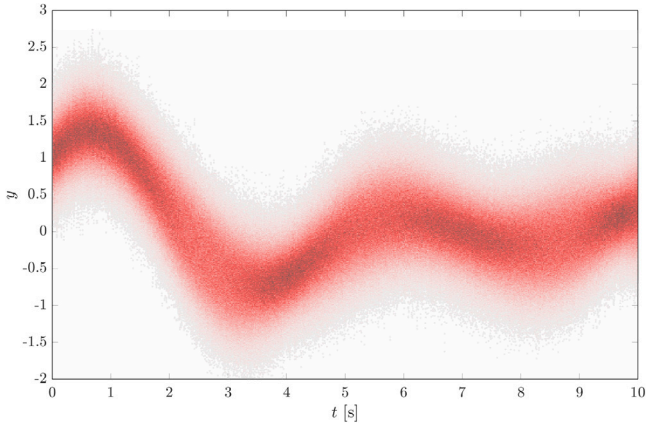

Fig. 3. Realizations of response variables in the original space. *The color-map represent the density of the sample points.*

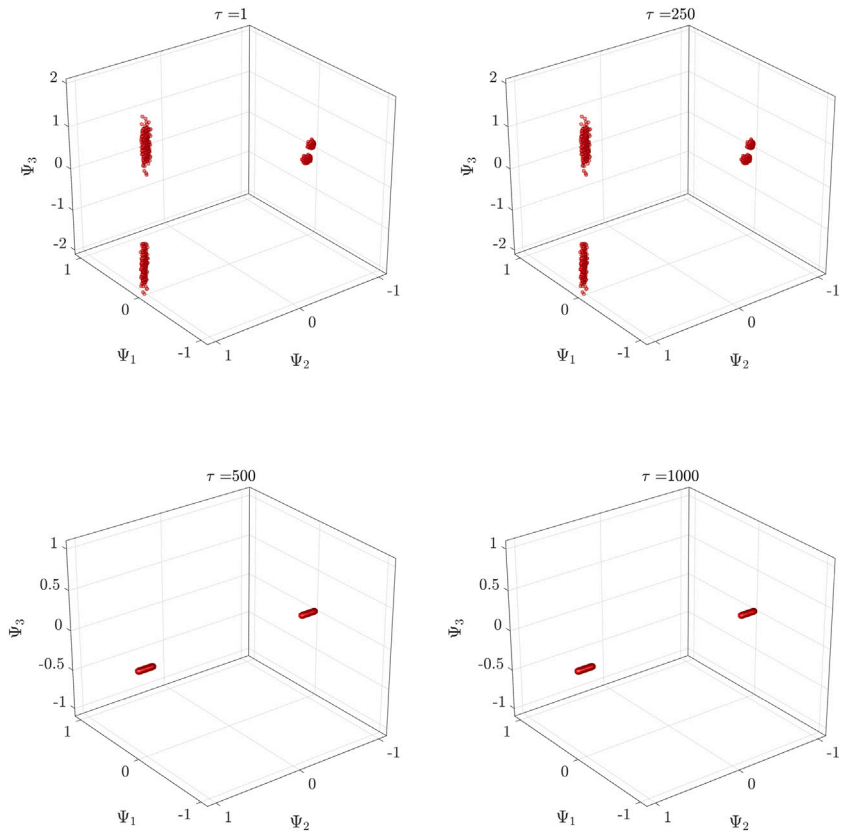

Fig. 4. Feature space representation using the diffusion map with various time-scales.

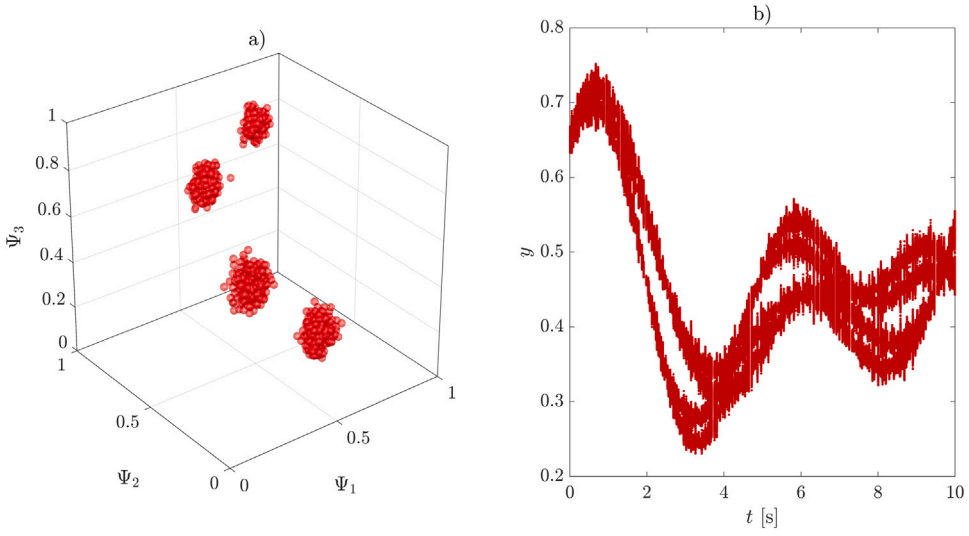

Fig. 5. Feature space representation (a) and reconstruction (b) using the autoencoder.

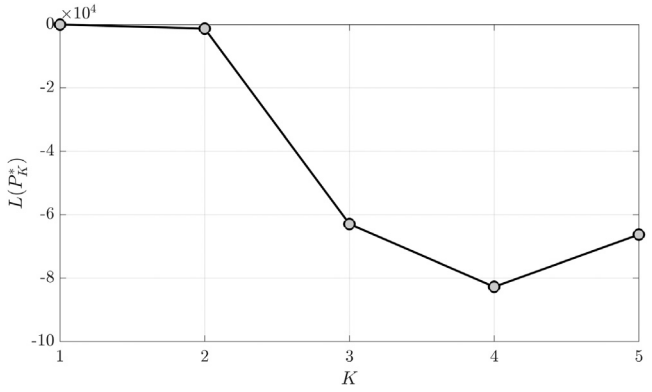

Fig. 6. The $\ell(P_K^*)$-K curve of the information theoretic approach.

employed for feature mapping. The sigmoid transfer function is employed for all neurons. The neural network is trained using the scaled conjugate gradient algorithm [91], with a mean square error cost function without sparsity or other regularization terms. Prior to training, a min–max normalization is applied to samples of $\boldsymbol{Y}$, since the output of a sigmoid function lies in $[0,1]$. Fig. 5 shows the 3-dimensional feature space representation and the reconstructed $\boldsymbol{Y}$. To obtain the reconstruction, an inverse of the min–max normalization is applied to the output layer.

Although one could visually identify the number of patterns, for illustrative purpose, the information theoretic approach is applied to the diffusion map of $\tau = 1$. Fig. 6 shows the $\ell(P_K^*)$-K curve obtained from the information theoretic approach. It is seen from the figure that there is an abrupt drop in $K = 4$, indicating a significant decrease in the distortion from grouping into three patterns to grouping into four patterns, thus suggesting $K^* = 4$.

Guided by the feature mapping, Fig. 7 shows the mean vectors of the four performance patterns obtained from a k-means clustering, compared with the deterministic part of Eq. (25). Fig. 8 shows the samples of $\boldsymbol{Y}$ corresponding to each performance pattern. It can be seen from Fig. 7 that the mean vectors fully capture the deterministic component of Eq. (25). It can also be observed from Figs. 7 and 8 that Pattern 1 is only slightly different from Pattern 3, and Pattern 2 is only slightly different from Pattern 4, while the difference between Pattern 1/Pattern 3 and Pattern 2/Pattern 4 is significant. This observation is consistent with the conclusion implied from multiple time-scale diffusion maps. As expected, the participation factors of each pattern are found to be around $1/4$.

One can obtain a parametrization of $f_{\boldsymbol{Y}}(\boldsymbol{y})$ by a Gaussian mixture model with four components. The mean vectors of the Gaussian components can be set to the vectors in Fig. 7, the covariance matrices are close to identity matrices, and each component weight is close to $1/4$. The generating densities can be parameterized by

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = f_{\boldsymbol{X}}(\boldsymbol{x}_p, x_{st}) = \sum_{k=1}^{4} \lambda_k \delta(x_{st} - k) f_{\boldsymbol{X}}(\boldsymbol{x}_p | x_{st} = k), \tag{26}$$

where $f_{\boldsymbol{X}}(\boldsymbol{x}_p | x_{st} = k)$ can be parameterized by Gaussian distributions. The Dirac function appears in Eq. (26) because there is a discrete random variable.

Finally, it is of interest to consider the case that X_{st} cannot be observed. In this case, the generating densities in the space of $\boldsymbol{X}_p$ parameterized by a Gaussian mixture model are devoid of identifiability, i.e., each Gaussian component in the mixture cannot be differentiated from the others. This is because $\boldsymbol{X}_p$ merely adds random noises to the output (see Eq. (25)), and the identifiability of $\boldsymbol{X}$ comes from the X_{st} component. However, knowing the fact that the generating densities lack identifiability is a meaningful observation, since this implies there are missing basic random variables or the performance patterns stem from deterministic mechanisms.

6.2. A stochastic Lorenz system

Consider a Lorenz system described by the following ordinary differential equations [92],

$$\begin{aligned} \frac{d\,y_1}{d\,t} &= \sigma(y_2 - y_1) \\ \frac{d\,y_2}{d\,t} &= y_1(\rho - y_3) - y_2 \\ \frac{d\,y_3}{d\,t} &= y_1 y_2 - \beta y_3 \end{aligned} \tag{27}$$

where σ, ρ, and β are system parameters. Lorenz system was originally developed to model convection rolls in the atmosphere, but it can also be used to describe the motion of mechanical systems, e.g., Lorenz Waterwheel [93]. In this example, we set $\sigma = 10$, $\beta = 8/3$, and ρ to be a Gaussian random variable with mean 24 and variance 1. The initial condition of Eq. (27), $[y_1(0), y_2(0)]$, is set to be a bi-variate Gaussian random variable with zero mean and identity covariance matrix, while $y_3(0)$ is fixed to zero. In the context of PPPD, the basic random variables, $\boldsymbol{X}$, are $X = [\rho, y_1(0), y_2(0)]$, and $\boldsymbol{X}$ is multivariate Gaussian with mean $[24, 0, 0]$ and identity covariance matrix. The response variables, $\boldsymbol{Y}$, are discretized random processes describing the time evolution of $[y_1, y_2, y_3]$, i.e., $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3]$, and $\boldsymbol{y}_j = [y_j(t_1), \ldots, y_j(t_n)]$, $j = 1, 2, 3$. It is assumed the whole sample space of $\boldsymbol{Y}$ is of interest. The Lorenz system is simulated from time 0 to 100, with a uniform incremental time step of 0.01. Therefore, the dimension of $\boldsymbol{Y}$ is $3 \times (100/0.01 + 1) = 30003$.

Using a direct Monte Carlo simulation and Runge–Kutta method, 2000 random realizations (shown in Fig. 9) of $\boldsymbol{Y}$ are obtained.

Fig. 10 shows realizations of $\boldsymbol{Y}$ embedded into a 3-dimensional feature space, obtained from the diffusion map (with identical settings as that in the previous example), and a 5 hidden layer autoencoder (with identical settings as that in the previous example). For the application of diffusion map in this example, we do not observe qualitatively different behaviors by varying time-scale τ in a relatively wide range, thus only the diffusion map with $\tau = 1$ is illustrated.

The reconstructed $\boldsymbol{Y}$ from the autoencoder is shown in Fig. 11. It can be observed from the feature space representation that: (a) the random trajectories of the Lorenz system can be classified into four patterns; and (b) the four patterns can be further divided into two groups, in one group the samples are tightly clustered while in the other the samples are dispersed. According to properties of Lorenz systems, it is reasonable to conjecture that the aforementioned four patterns are associated with periodic trajectories (where there are two *attractors*) and chaotic trajectories (where there are two *repellors*). Incidentally, one may observe from Fig. 11 that the reconstructed trajectories are in low accuracy. However, in the context of this study, as long as the main features are captured, the reconstruction quality of the autoencoder is not of much practical importance.

Fig. 12 shows characteristic trajectories of the four patterns obtained from a "hierarchical density-based spatial clustering of applications with noise (HDBSCAN)" [94] clustering. Fig. 13 shows the samples of $\boldsymbol{Y}$ corresponding to each pattern. The HDBSCAN instead of the simple k-means clustering is used here since HDBSCAN performs better when handling dataset with varying shapes and densities. The characteristic trajectory for each pattern is obtained as the sample closest to the cluster mean.

It can be observed from Figs. 12 and 13 that Pattern 1/Pattern 2 correspond to periodic trajectories in which the system eventually oscillates around one of the two attractors, while Pattern 3/Pattern 4 correspond to chaotic trajectories in which the system is repelled by the two repellors. The participation factors of the four patterns are

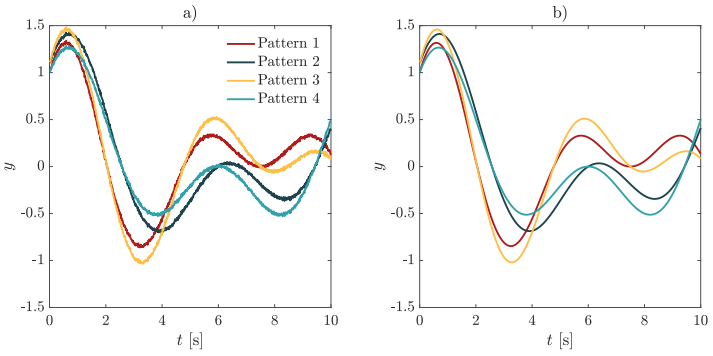

Fig. 7. Mean vectors of four patterns of $\boldsymbol{Y}$ (left) compared with the deterministic part of Eq. (25) (right).

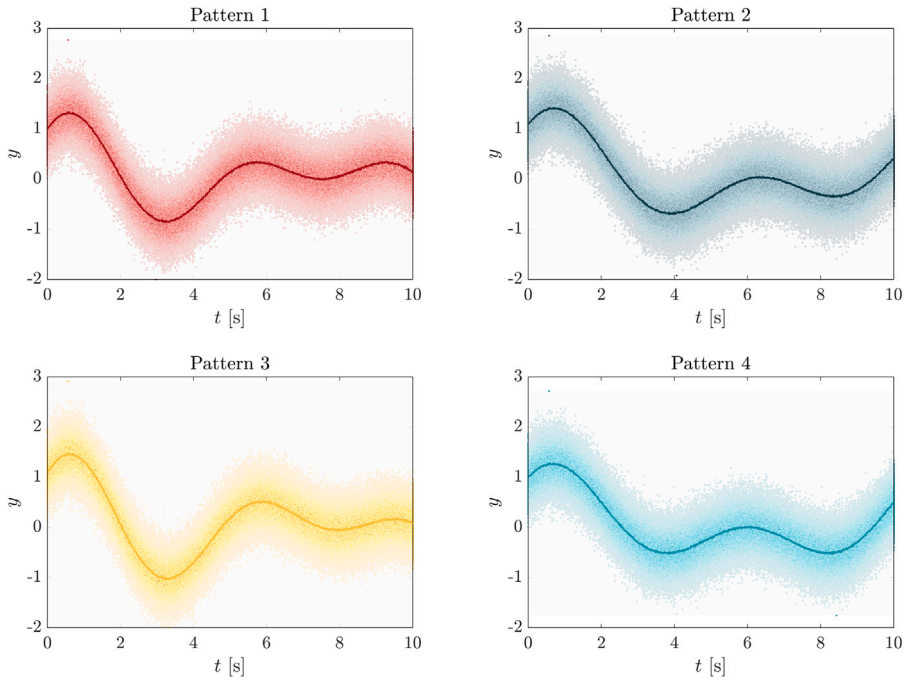

Fig. 8. Samples of $\boldsymbol{Y}$ corresponding to each pattern.

estimated as 0.267, 0.256, 0.243 and 0.234, for Pattern 1, 2, 3 and 4, respectively.

Next, Fig. 14 illustrates how the patterns are triggered in the sample space of basic random variables, $\boldsymbol{X} = [\rho, y_1(0), y_2(0)]$. One can observe a clear boundary in the $y_1(0)$-$y_2(0)$ plane that separates Pattern 1/Pattern 4 from Pattern 2/Pattern 3. This is because the initial trajectories (trajectories near the initial state) for Pattern 1/Pattern 4 (or Pattern 2/Pattern 3) are similar and they are controlled by the initial condition $[y_1(0), y_2(0)]$. One can also see that for relatively large ρ values, the Lorenz system is chaotic, and for relatively small ρ values the system is periodic. In fact, the smallest ρ value for samples in Pattern 3/Pattern 4 is 24.09, which is fairly close to the theoretical criticality $\rho^* = 24.06$.[6]

[6] A critical ρ of 24.06 means that a *strange attractor* corresponds to chaotic trajectories appears at $\rho > 24.06$.

6.3. An earthquake engineering example

Consider a 3-story shear-building model shown in Fig. 15. The building model is subjected to stochastic ground motion excitation. The force–deformation behavior of each column is assumed to be linearly elastic. The stiffness of each column, k_1, k_2, and k_3, independently follows a log-normal distribution with mean 6.0×10^7 [N/m] and coefficient of variation (c.o.v.) of 0.05. The floor masses are identical and equal to 3×10^4 [kg], and 5% damping ratio is assumed for each mode. The building is subjected to a stochastic ground motion with the auto power spectrum density (PSD) described by a modified Kanai–Tajimi model suggested by Clough and Penzien [95],

$$S_f(\omega) = S_0 \frac{\omega_f^4 + 4\zeta_f^2\omega_f^2\omega^2}{(\omega_f^2 - \omega^2)^2 + 4\zeta_f^2\omega_f^2\omega^2} \frac{1}{(\omega_s^2 - \omega^2)^2 + 4\zeta_s^2\omega_s^2\omega^2}, \tag{28}$$

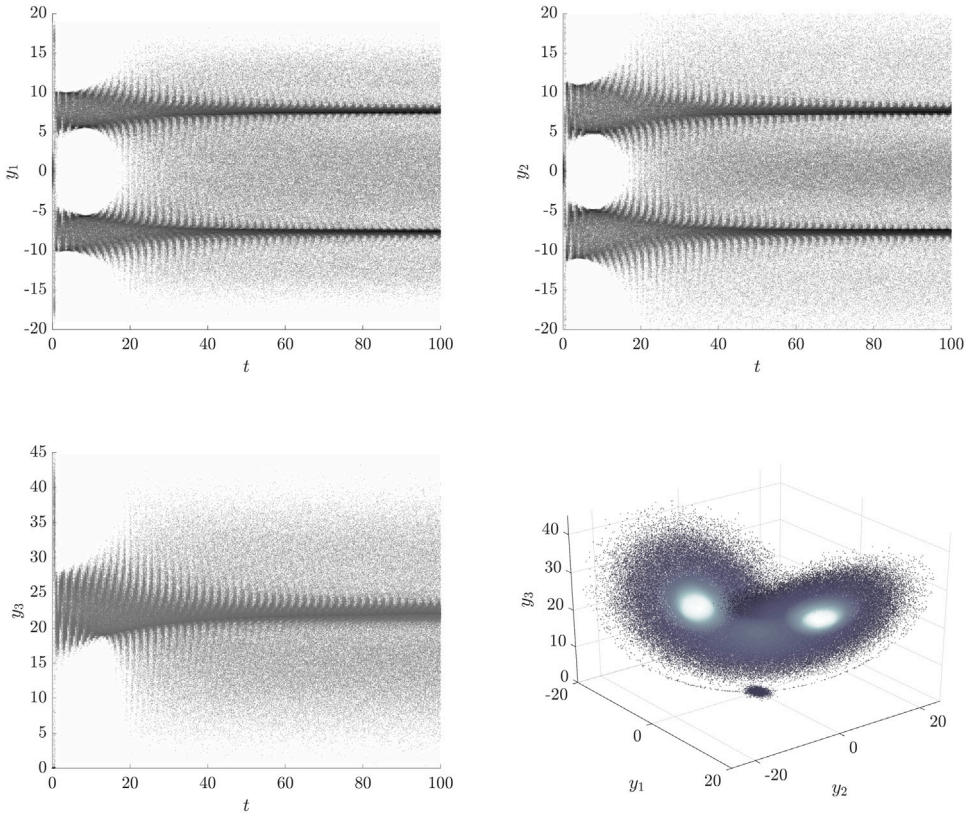

Fig. 9. Realizations of $\boldsymbol{Y}$ in the original space.

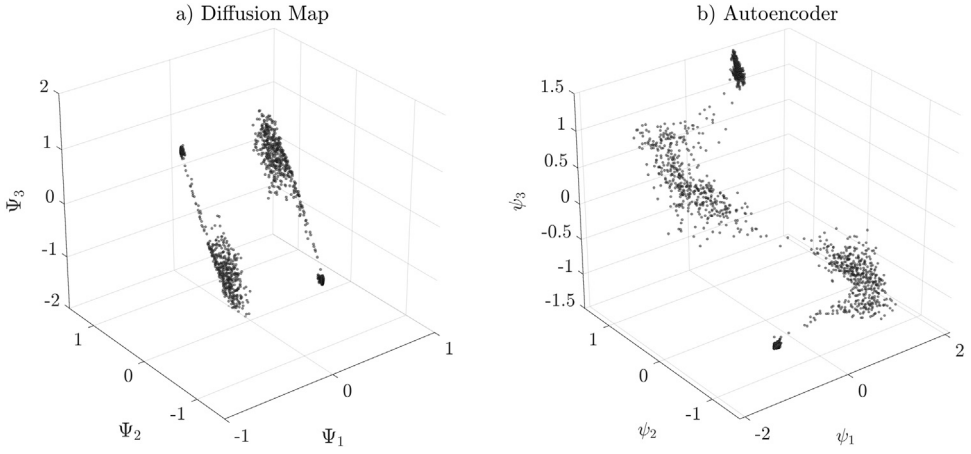

Fig. 10. Feature space representation using the diffusion map and the autoencoder.

where $S_0 = 0.0015$ [m^2/s^3] is a scale factor, $\omega_f = 15$ [rad/s] and $\zeta_f = 0.6$ are the filter parameters representing, respectively, the natural frequency and damping ratio of the soil layer, and $\omega_s = 0.5$ [rad/s] and $\zeta_s = 0.6$ are parameters of a second filter that is introduced to assure finite variance of the ground displacement. The duration of the ground motion is assumed to be 10 s.

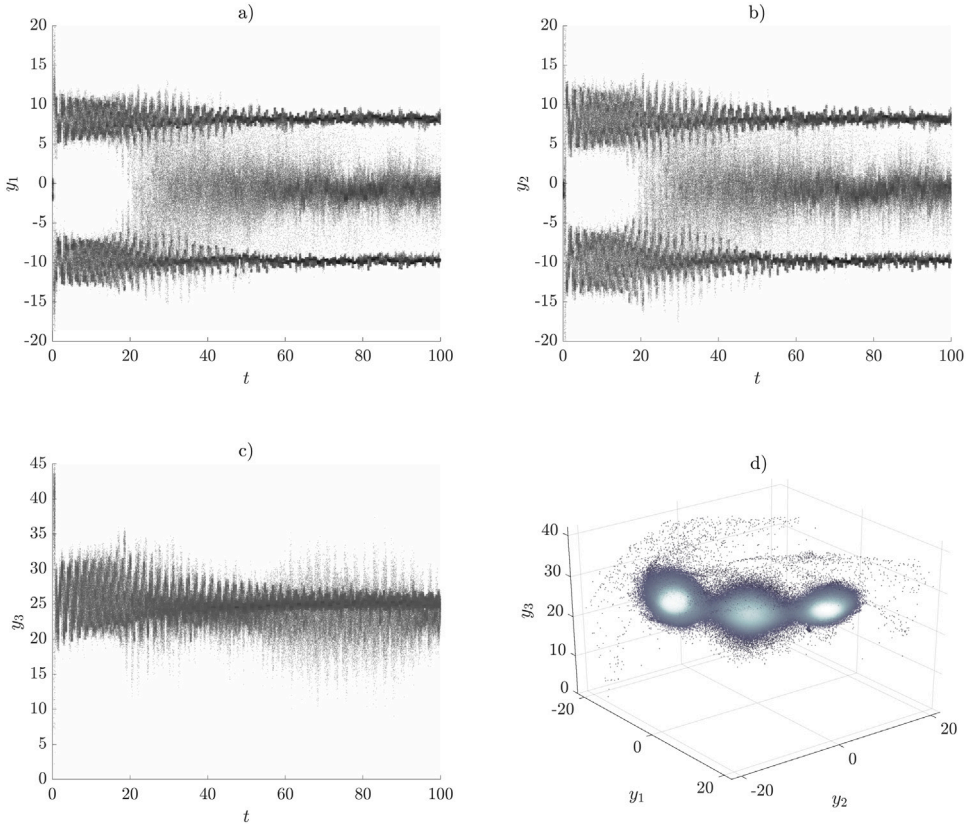

Fig. 11. Reconstruction using the autoencoder.

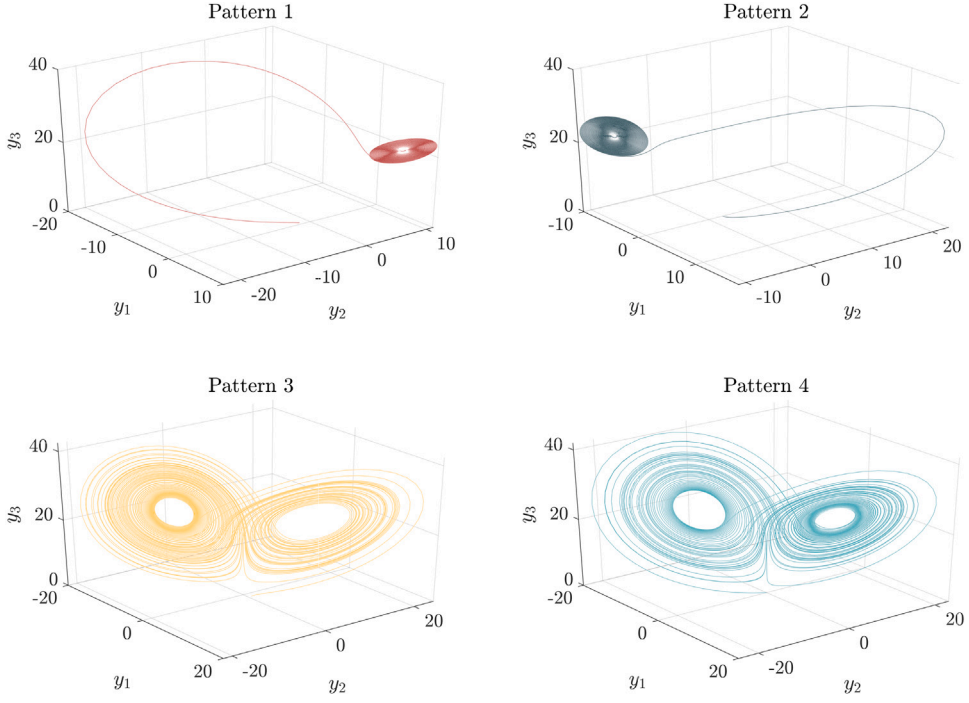

Fig. 12. Characteristic trajectories for each pattern.

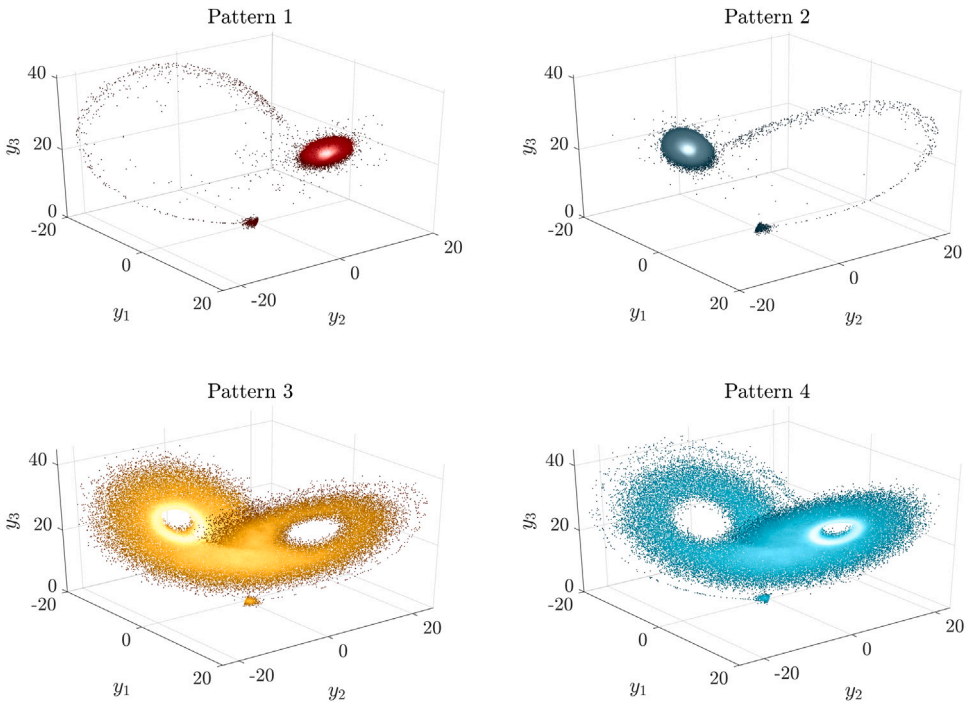

Fig. 13. Sample trajectories corresponding to each pattern.

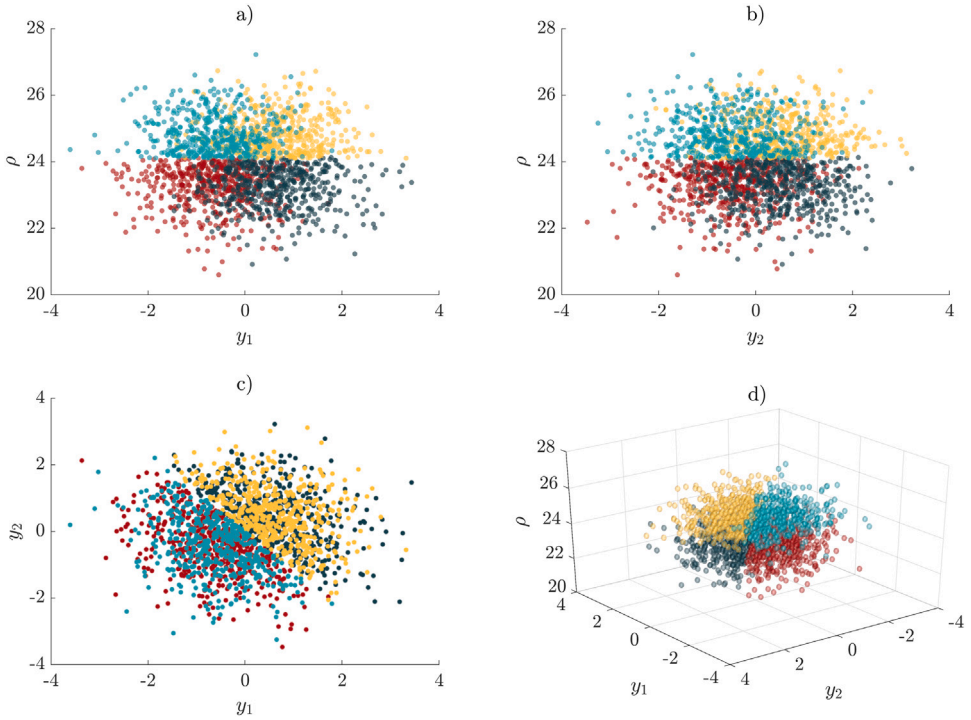

Fig. 14. Realizations of basic random variables corresponding to each pattern.

The stochastic ground motion process $X_g(t)$ is discretized in frequency domain as [96]

$$X_g(t) = \sum_{j=1}^{p/2} \sigma(\omega_j)(x_j \cos(\omega_j t) + x'_j \sin(\omega_j t)), \tag{29}$$

where x_j, x'_j are independent standard Gaussian variables, the frequency point is given by $\omega_j = j\Delta\omega$ with $p/2 = 200$, the cut-off frequency is set to $\omega_{p/2} = 15\pi$ (therefore $\Delta\omega = 30\pi/p = 0.075\pi$), and $\sigma(\omega_j) = \sqrt{2S_f(\omega_j)\Delta\omega}$.

Given the specifications of the stochastic process $X_g(t)$, the set of basic random variables, $\boldsymbol{X}$, can be written as $\boldsymbol{X} = [x_1, x'_1, \ldots, x_{200}, x'_{200}, k_1, k_2, k_3]$, and the dimension of $\boldsymbol{X}$ is $400 + 3$ (400 for ground motion and 3 for random stiffness). The response variables $\boldsymbol{Y}$ are discretized random processes describing the time evolution of each inter-story displacement, i.e., relative displacement between roof and ground for each story, and is written as $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{y}_3]$, and $\boldsymbol{y}_j = [y_j(t_1), \ldots, y_j(t_n)]$,

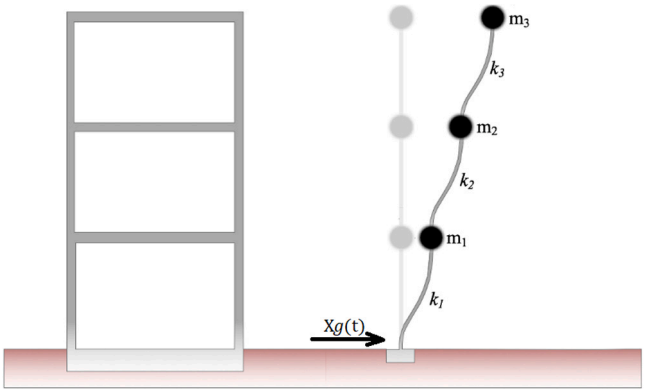

Fig. 15. Shear-building model.

$j = 1, 2, 3$. The shear-building model is simulated from 0 to 10 s, with a uniform incremental time step of 0.01. Therefore, the dimension of $\boldsymbol{Y}$ is $3 \times (10/0.01 + 1) = 3003$. We are interested in the performance state defined by

$$\mathcal{P}_{\boldsymbol{Y}} = \{\boldsymbol{y} | c - \max |\boldsymbol{y}| \leq 0\} , \tag{30}$$

where c is a threshold value for the inter-story displacement.

Using a Hamiltonian Monte Carlo based subset simulation (see Appendix A and [97]), for threshold values $c = 0.02$ [m] and $c = 0.12$ [m], we obtain 5000 random realizations (shown in Fig. 16). The probabilities of $\boldsymbol{Y} \in \mathcal{P}_{\boldsymbol{Y}}$ for threshold values $c = 0.02$ [m] and $c = 0.12$ [m] are estimated as 3.4×10^{-2} and 1.2×10^{-7}, respectively. It is seen from Fig. 16 that for larger threshold, the response of story 3 is surprisingly smaller. This phenomenon can be qualitatively understood as: the system has to find "efficient" route to trigger the failure event, and it is not efficient for the story 3 to exceed a high response value. As the threshold value increases, the possibility that story 3 exceeds a high response value is ruled out in a natural selection manner. The following discussions in this section will provide further evidence to support the aforementioned qualitative explanation.

Fig. 17 shows realizations of $\boldsymbol{Y}$ embedded into a 3-dimensional feature space, obtained from the diffusion map (with identical settings as that in the previous examples). It can be observed from the figure that as the threshold increases, the number of performance patterns changes from 3 to 2.

For the two thresholds, pattern identification analysis with k-means clustering is performed. Figs. 18 and 19 show characteristic trajectories of the performance patterns. The characteristic trajectory for each pattern is obtained as the sample closest to the cluster mean. The participation factors of each pattern for threshold $c = 0.02$ [m] are estimated as 0.66, 0.27, and 0.07 for Pattern 1,2 and 3, respectively, while the participation factors for threshold $c = 0.12$ [m] are 0.82 and 0.18 for Pattern 1 and 2, respectively.

The following remarks can be made on the performance patterns.

(a) For threshold $c = 0.02$ [m], the three patterns correspond to the degree of dominance of each inter story displacement. In Pattern 1, the first inter story displacement is in general larger than that of the other two stories; in Pattern 2, the second inter story displacement dominates; and in Pattern 3, the third inter story displacement dominates.

(b) For threshold $c = 0.02$ [m], the participation factors for each pattern (0.66, 0.27, and 0.07) indicate that it is most likely to have the first inter story displacement larger than the other stories, and it is least likely that the third inter story displacement dominates. This observation can be explained by: the inertia force applied to the first story is the largest, and the mean stiffness of each story is identical.

(c) For threshold $c = 0.12$ [m], the Pattern 3 in threshold $c = 0.02$ [m] disappears,[7] and the Pattern 1 and Pattern 2 are retained. Given this trend, it is reasonable to conjecture that if the threshold is set higher, only one pattern (the scenario that the first story displacement dominates) would remain. This conjecture is confirmed by performing PPPD analysis for $c = 0.15$ [m]. The failure probability for $c = 0.15$ [m] is estimated as 9.9×10^{-9}, and the feature space representation is given by Fig. 20. It can be seen from Fig. 20 that all points cluster together, suggesting there is only one pattern.

(d) From threshold $c = 0.02$ to $c = 0.12$, the typical performance pattern trajectories, in general, have a frequency shift to the relatively low frequency side.

Next, we investigate how the performance patterns are generated by basic random variables. To start with, we study the relationship between performance patterns and ground motions. To obtain a clear illustration, instead of showing samples of $\boldsymbol{X}$, we estimate the power spectrum density (PSD) of ground motion samples corresponding to each pattern, and the results are shown in Fig. 21. Recall that the ground motion is a deterministic function (Eq. (29)) of $\boldsymbol{X}$. The analytical auto-PSD model of the ground motion (Eq. (28)) is also shown in the figure for a comparison. It is seen from the figure that at each threshold, the PSD for each pattern looks similar. Therefore, we conclude that **for a given threshold**, the performance patterns are not generated by ground motions with different characteristics. However, it is important to observe that this conclusion does not suggest the frequency contents of ground motion do not influence the performance patterns. In fact, it can be observed that the PSD of each performance pattern for threshold $c = 0.12$ [m] has richer low frequency contents than that for threshold $c = 0.02$ [m].

Given the aforementioned investigation, we conjecture that in the space of random stiffnesses $[k_1, k_2, k_3]$, there should be clear patterns. This assumption is confirmed by Fig. 22, which shows realizations of $[k_1, k_2, k_3]$ corresponding to each performance pattern. Fig. 22 provides a way to design/control the stochastic behavior of the building, so that the random first passage event of maximum responses can be manipulated. Note that in this example, Fig. 22 can be anticipated, because to have a high likelihood of first passage in certain story, the stiffness at that story should be relatively small. However, PPPD analysis provides the quantitative approach to estimate the most likely setting that triggers certain performance pattern in a rare event.

Finally, it is important to highlight that the results of PPPD are obtained without a knowledge on the underlying physical/mathematical laws that govern the stochastic system. For all the examples studied in this paper, we use the governing laws to generate random samples, however, if the samples are obtained by performing real experiments or collecting data from sensors, the PPPD analysis can be applied in the same manner. This perspective further highlights the potential applications of PPPD analysis.

7. Conclusions

A framework termed Probabilistic Performance-Pattern Decomposition (PPPD) is developed to facilitate an in-depth understanding on the behaviors of stochastic systems. The concept of performance-pattern is developed using response variables, which directly describe the behavior of a stochastic system, and the performance state, which is a specified subset in the sample space of response variables. The theoretical framework of PPPD is proposed via a probabilistic decomposition of response variables conditional on the performance state. The computational framework of PPPD is consisted of three major ingredients:

[7] Rigorously speaking, the possibility of having Pattern 3 is extremely small, approximated as $1.2 \times 10^{-7} \times 1/5000 \approx 2.4 \times 10^{-11}$. Recall that we have simulated 5000 samples in the performance state.

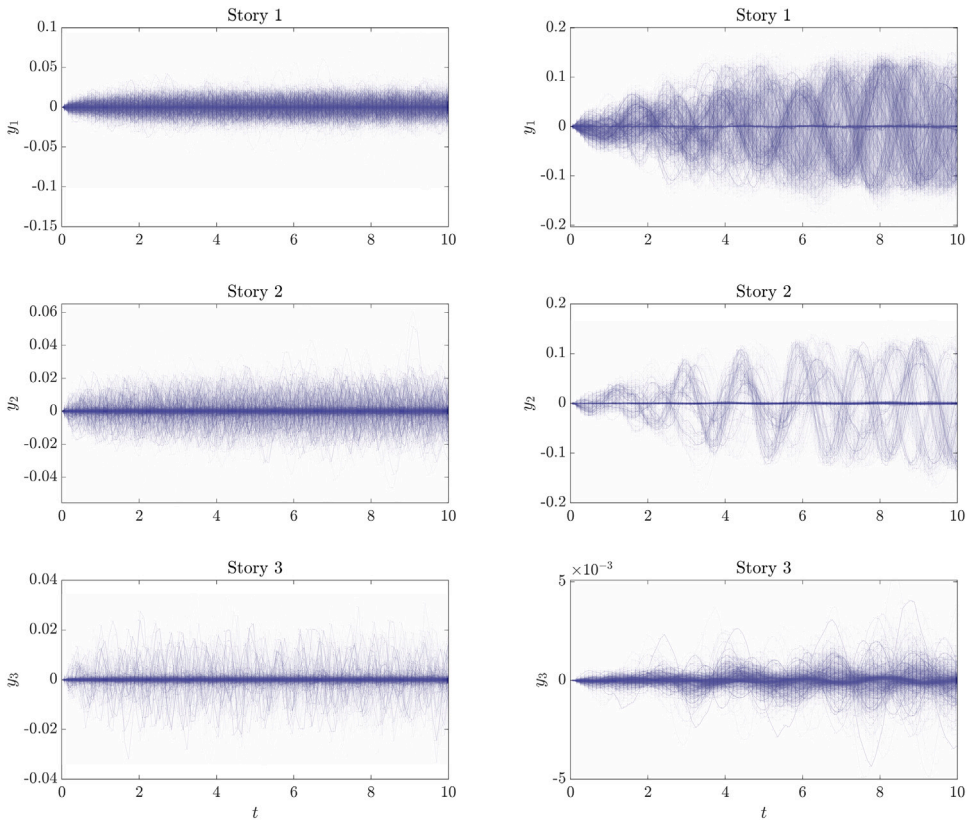

Fig. 16. Realizations of $\boldsymbol{Y}$ in the original space for thresholds $c = 0.02$ [m] (left) and $c = 0.12$ [m] (right).

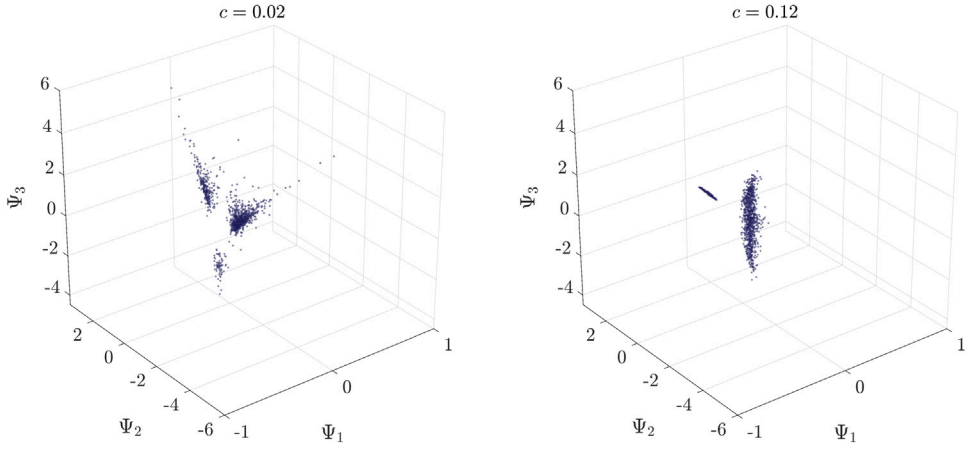

Fig. 17. Feature space representation using the diffusion map for thresholds $c = 0.02$ [m] (left) and $c = 0.12$ [m] (right).

(1) event observation; (2) feature mapping; and (3) pattern identification. Using rare event simulation, manifold learning and clustering techniques, the computational framework of PPPD is capable of analyzing complex stochastic systems involving random fields/processes, and producing the main behavior patterns of the system conditional on the performance state of interest. Moreover, PPPD analysis enables the identification of critical domains in the sample space of basic random variables that trigger each performance pattern.

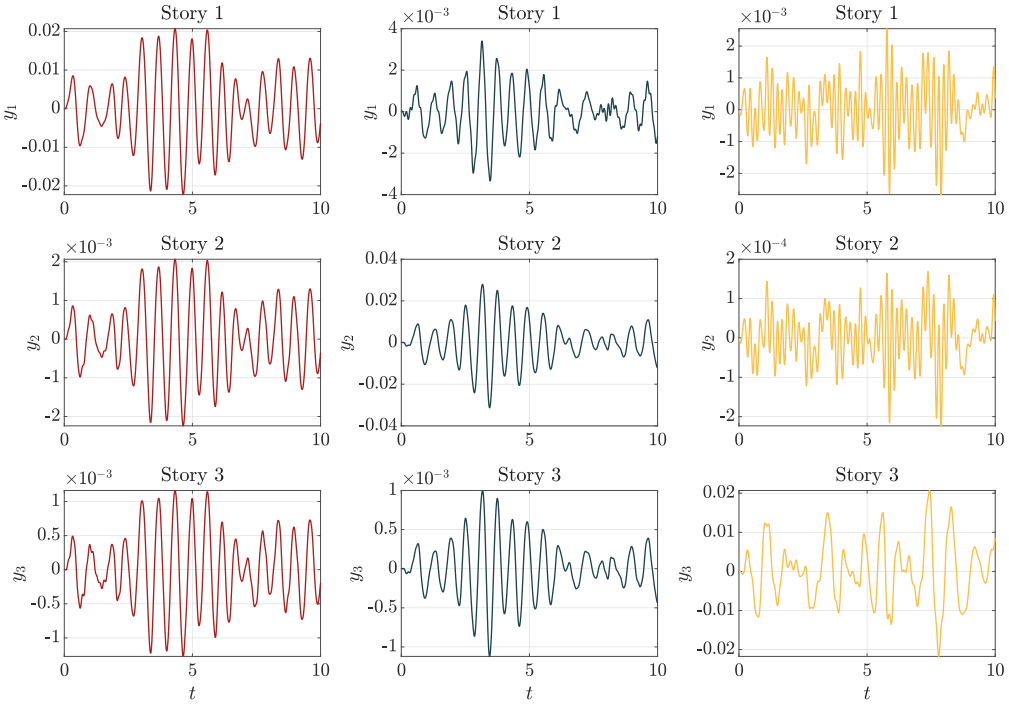

Fig. 18. Characteristic trajectories for each pattern (threshold $c = 0.02$ [m]). *The left, middle, and right column of plots show the Pattern 1, Pattern 2, and Pattern 3, respectively*.

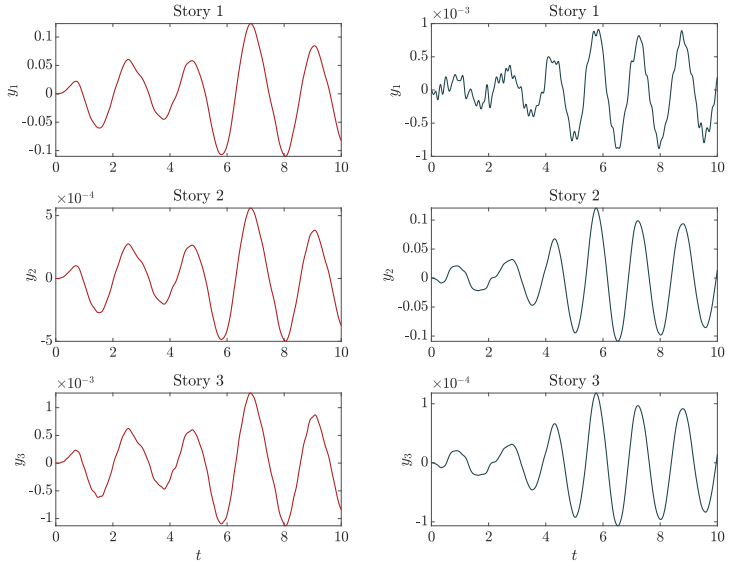

Fig. 19. Characteristic trajectories for each pattern (threshold $c = 0.12$ [m]). *The left and right column of plots show the Pattern 1 and Pattern 2, respectively*.

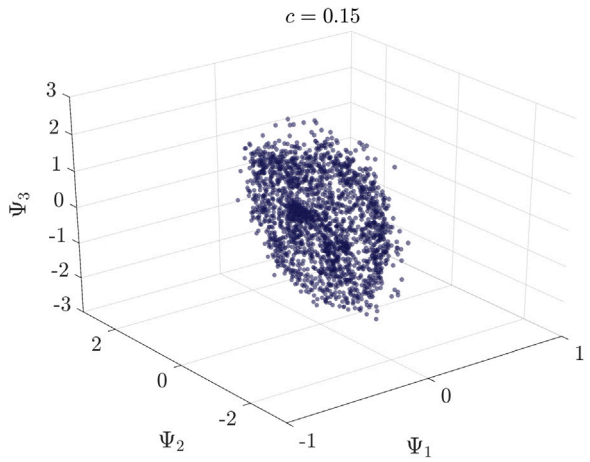

Fig. 20. Feature space representation using the diffusion map for threshold $c = 0.15$ [m].

To illustrate the effectiveness of PPPD, the paper investigates three numerical examples, which all involve random processes and high-dimensional input uncertainties. The first example is a hypothetical system with analytical stochastic input and output processes. A PPPD analysis for this example results in four performance patterns, which are consistent with mathematical rules of the hypothetical system. The second example is a Lorenz system with random system parameters and initial conditions. The PPPD analysis identifies periodic and chaotic response trajectories, and quantitatively predicts how different performance patterns are generated. The last example is a simplified shear-building model with random stiffnesses and subjected to a stochastic ground motion excitation described by a power spectrum density model. A PPPD analysis for this example leads to insightful results on how the performance patterns shift with the decrease of failure probability, and how the patterns are generated in the space of basic random variables. A promising application of PPPD is to use it in a fully data-driven fashion to discover patterns and regularities of large-scale stochastic systems. Ultimately, PPPD can assist the physics-informed decision process.

CRediT authorship contribution statement

Ziqi Wang: Writing – review & editing, Writing – original draft, Validation, Software, Formal analysis, Data curation, Conceptualization. **Junho Song:** Writing – review & editing, Writing – original draft, Conceptualization. **Marco Broccardo:** Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work of J. Song was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean government (NRF-2021R1A2C2003553) and the Institute of Construction and Environmental Engineering at Seoul National University. This support is gratefully acknowledged. Prof. Marco Broccardo has been supported by the Italian Ministry of Education, University and Research (MIUR) in the frame of the "Departments of Excellence 2023–2027 (grant L232/2016)". Any opinions, findings, and conclusions expressed in this paper are those of the authors, and do not necessarily reflect the views of the sponsors.

Appendix A. Sequential Monte Carlo to sample from the performance state

Algorithm 2 Sequential Monte Carlo simulation to generate N random realizations from $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$

Step 1: Parameter specification

- Define p_0, the conditional probability for each intermediate states.
- Define N_0, the sample size in each intermediate step. We let $N_0 \cdot p_0 \approx N$.

Step 2: Initial run

- Draw N_0 samples, $\boldsymbol{x}_0^{(i)}$, $i = 1, 2, ..., N_0$, from $f_{\boldsymbol{X}}(\boldsymbol{x})$.
- Evaluate $\boldsymbol{y}_0^{(i)} = \mathcal{M}(\boldsymbol{x}_0^{(i)})$ and $G(\boldsymbol{y}_0^{(i)})$, $i = 1, 2, ..., N_0$.
- Sort samples $\boldsymbol{x}_0^{(i)}$ and $\boldsymbol{y}_0^{(i)}$ in increasing orders of $G(\boldsymbol{y}_0^{(i)})$.
- Find $g^{(1)}$ as the p_0 percentile of $G(\boldsymbol{y}_0^{(i)})$, so that $\mathcal{P}_{\boldsymbol{X}}^{(1)}$ is specified as $\mathcal{P}_{\boldsymbol{X}}^{(1)} = \{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} : G(\mathcal{M}(\boldsymbol{x})) - g^{(1)} \leq 0\}$.
- Set $j \leftarrow 1$.

Step 3: Iterative runs

- Repeat while $g^{(j)} > 0$
  - Starting from $p_0 \cdot N_0$ seed samples $\boldsymbol{x}_{j-1}^{(i)}$, $i = 1, ..., p_0 \cdot N_0$, that have $\boldsymbol{x}_{j-1}^{(i)} \in \mathcal{P}_{\boldsymbol{X}}^{(j)}$, use a MCMC sampler to drawn $(1 - p_0)N_0$ samples from $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}}^{(j)})$.
  - Store the $p_0 \cdot N_0 + (1 - p_0)N_0 = N_0$ samples that lie in $\mathcal{P}_{\boldsymbol{X}}^{(j)}$ as $\boldsymbol{x}_j^{(i)}$.
  - Sort samples $\boldsymbol{x}_j^{(i)}$ in increasing orders of $G(\boldsymbol{y}_j^{(i)})$, where $\boldsymbol{y}_j^{(i)} = \mathcal{M}(\boldsymbol{x}_j^{(i)})$.
  - Find $g^{(j+1)}$ as the p_0 percentile of $G(\boldsymbol{y}_j^{(i)})$, so that $\mathcal{P}_{\boldsymbol{X}}^{(j+1)} = \{\boldsymbol{x} \in \mathbb{R}^{|\boldsymbol{X}|} : G(\mathcal{M}(\boldsymbol{x})) - g^{(j+1)} \leq 0\}$.
  - Set $j \leftarrow j + 1$.

Step 4: Final MCMC sampling

- Use all samples in $\mathcal{P}_{\boldsymbol{X}}$ as seeds, perform MCMC sampling until a total of N samples in $\mathcal{P}_{\boldsymbol{X}}$ are obtained.

To have a representative set of realizations of $f_{\boldsymbol{X}}(\boldsymbol{x}|\mathcal{P}_{\boldsymbol{X}})$, the MCMC algorithm used in Algorithm 2 should be effective. The Hamiltonian Monte Carlo (HMC) method has proven to be highly effective in this context [98]. Implementation details of the HMC algorithm in the context of subset simulation can be found in [97].

Appendix B. Implementation of diffusion map for PPPD

For the dataset $\mathcal{Y}$, the basic procedures of constructing feature vectors $\boldsymbol{\Psi}$ using the diffusion map is described as follows.

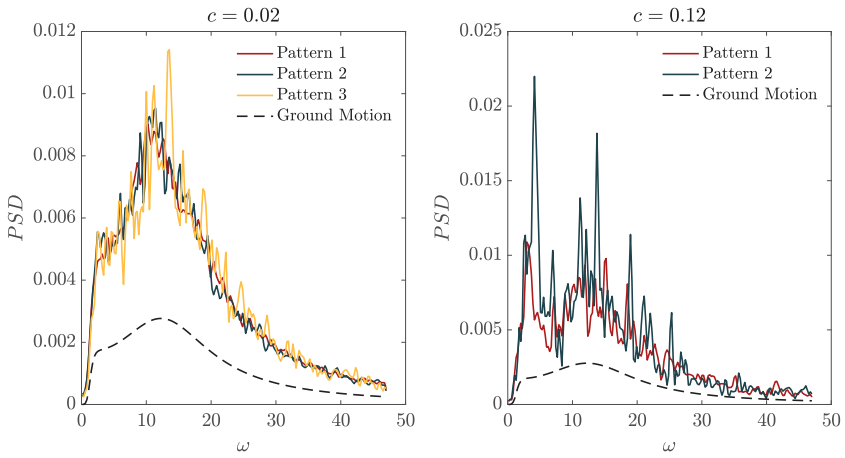

Fig. 21. Power spectrum density for ground motion samples of each performance pattern.

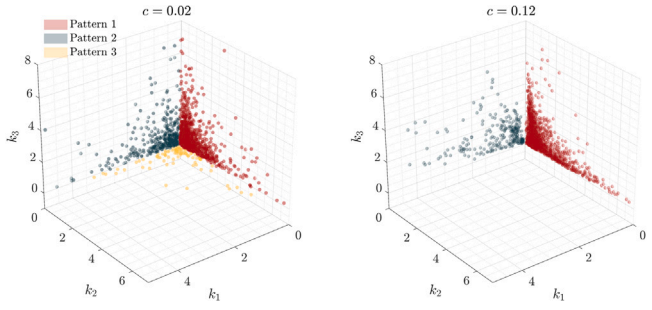

Fig. 22. Realizations of random stiffnesses corresponding to each pattern.

Algorithm 3 Constructing feature vectors $\boldsymbol{\Psi}$ from $\mathcal{Y}$ using the diffusion map

Step 1: Construct the similarity matrix

- Construct the similarity matrix $\boldsymbol{W} = \{w_{ij}\}$, $i, j = 1, 2, ..., N$, where $w_{ij} = s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$, $s(\cdot)$ is a specified similarity function.

Step 2: Obtain the Markov matrix

- Normalize $\boldsymbol{W}$ by $\widehat{\boldsymbol{W}} = \boldsymbol{D}^{-\alpha}\boldsymbol{W}\boldsymbol{D}^{-\alpha}$, where $\boldsymbol{D}$ is a diagonal matrix with $\boldsymbol{D}_{ii} = \sum_{j=1}^{N} w_{ij}$, and α, $\alpha \in \mathbb{R}$, is a specified parameter.
- Compute the Markov matrix $\boldsymbol{M}$ by $\boldsymbol{M} = \widehat{\boldsymbol{D}}^{-1}\widehat{\boldsymbol{W}}$, where $\widehat{\boldsymbol{D}}$ is a diagonal matrix with $\widehat{\boldsymbol{D}}_{ii} = \sum_{j=1}^{N} \hat{w}_{ij}$.

Step 3: Obtain the feature vectors

- Compute the n_t largest eigenvalues and the corresponding eigenvectors for matrix $\boldsymbol{M}$, denoted by λ_i and ϕ_i, $i = 1, 2, ..., n_t$, respectively.
- Compute feature vectors $\boldsymbol{\Psi} = [\boldsymbol{\psi}^{(1)}, .., \boldsymbol{\psi}^{(N)}]$ by $\boldsymbol{\Psi} = \boldsymbol{\Lambda}^{\tau}\boldsymbol{\Phi}^{T}$, where Λ is a $n_t \times n_t$ diagonal matrix with $D_{ii} = \lambda_i$; τ, $\tau \in \mathbb{N}^{+}$, is a scale parameter describing the time-scale of the diffusion process; $\boldsymbol{\Phi}^{T}$ is the transpose of the $N \times n_t$ eigenmatrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{n_t}]$.

B.0.1. Remark 1 of Algorithm 3: selecting the similarity function

The similarity function $s(\cdot)$ could be problem-specific, but has to satisfy: (a) $s(\cdot)$ is symmetric, i.e., $s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) = s(\boldsymbol{y}^{(j)}, \boldsymbol{y}^{(i)})$, and (b) $s(\cdot)$ is positivity preserving, i.e., $s(\cdot) \geq 0$. A common choice of $s(\cdot)$ is of the exponential form written as:

$$s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)}) = \exp\left[-\frac{d^2(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})}{\epsilon}\right], \tag{B.1}$$

where $d(\cdot)$ is a specified distance function, and ϵ is a specified scale parameter. If $d(\cdot)$ is the Mahalanobis distance, Eq. (B.1) is equivalent to the Gaussian kernel.

In PPPD, if one has physical insight or other intuition on how radically different one realization of response variables is from another, it should be reflected in the definition of $d(\cdot)$. Otherwise, one may use conventional distance measures such as the L^p-norm. Note that if an L^p-norm distance is used in Eq. (B.1), the similarity matrix $\boldsymbol{W}$ will be dense, since every entry of $\boldsymbol{W}$ is nonzero. This would lead to storage and efficiency issues for a large dataset. A simple remedy is to convert entries of $\boldsymbol{W}$ with values below some threshold to zero. An alternative approach to obtain a sparse $\boldsymbol{W}$ is to use methods such as k-nearest neighbor (k-NN) [99] to determine $s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$. Specifically, $s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$ can be obtained via: if $\boldsymbol{y}^{(i)}$ is within the k-nearest neighbors of $\boldsymbol{y}^{(j)}$, *or* $\boldsymbol{y}^{(j)}$ is within the k-nearest neighbors of $\boldsymbol{y}^{(i)}$, where 'k-nearest' is measured by $d(\cdot)$, then $s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$ is computed from Eq. (B.1); otherwise, set $s(\boldsymbol{y}^{(i)}, \boldsymbol{y}^{(j)})$ to zero. A simple rule of thumb [99] to select the parameter "k" in k-NN algorithm is to set it in the order of $\log N$, where N is the number of samples in the dataset.

B.0.2. Remark 2 of Algorithm 3: selecting parameter α

The parameter α used in **Step 2** of the algorithm alters the amount of influence of sample densities over the eigen-functions and spectrum of the diffusion. It is analyzed in [83] that the parameter settings $\alpha = 0$, $\alpha = 0.5$, and $\alpha = 1$ are particularly meaningful. Specifically, $\alpha = 0$ corresponds to a Markov matrix that is identical to the random walk normalized Laplacian in graph theory, and the influence of the sample density is maximal; $\alpha = 0.5$ (approximately) corresponds to the diffusion of a Fokker–Planck equation; $\alpha = 1$ (approximately) corresponds to the Laplace–Beltrami operator (Brownian motion on the manifold where the data is sampled from), where one is able to recover the Riemannian geometry of the dataset. For the purpose of manifold learning, the setting of $\alpha = 1$ is suggested in many applications.

B.0.3. Remark 3 of Algorithm 3: selecting parameter τ

The parameter τ corresponds to the number of steps of running the Markov chain (characterized by the Markov matrix $\boldsymbol{M}$) forward in time. Therefore, instead of fixing τ, one could run Algorithm 3 for different time-scales to study the multiscale geometry of the dataset.

Appendix C. Implementation of autoencoder for PPPD

For the dataset $\mathcal{Y}$, the basic procedures of constructing feature vectors $\boldsymbol{\Psi}$ using the autoencoder is described as follows.

Algorithm 4 Constructing feature vectors $\boldsymbol{\Psi}$ from $\mathcal{Y}$ using the autoencoder

Step 1: Define architecture and parameters of the autoencoder

- Define the number of layers in the encoder/decoder, denoted as k_{cod}.
- Define the number of neurons in each layer of the encoder/decoder, denoted as n_j, $j = 1, ..., k_{cod}$.
- Define the activation functions for neurons in the encoder and decoder.
- Define the cost function for training the autoencoder.

Step 2: Perform layer-by-layer training

- Set dataset $D \leftarrow \mathcal{Y}$.
- Set $j \leftarrow 1$.
- Repeat while $j \leq k_{code}$
  - Set $n_{train} \leftarrow n_j$.
  - Using D as input, train an autoencoder with a single hidden layer of n_{train} neurons.
  - Set $\boldsymbol{\Psi}$ as the output of the hidden layer.
  - Set $D \leftarrow \boldsymbol{\Psi}$.
  - Set $j \leftarrow j + 1$.

Step 3: Fine-tuning the whole autoencoder (optional)

- Stack the single hidden layer autoencoders obtained in **Step 2** to form the whole deep autoencoder.
- Perform global fine-tuning for the whole autoencoder to optimize the reconstruction of $\mathcal{Y}$.
- Using $\mathcal{Y}$ as input for the tuned autoencoder, set $\boldsymbol{\Psi}$ as the output of the bottle-neck layer.

It is common practice to set the number of neurons in each hidden layer of the autoencoder to be smaller than the dimension of the input, otherwise there is a risk to learn the identity function. In general, it is common to start with a few layers and increase complexity as needed. Besides manipulating the network architecture, introducing a sparsity regularization term to the cost function is an alternative approach to enforce the learning of useful structures. In general, the cost function for an autoencoder can be of the form [100]:

$$cost = \frac{1}{N}\sum_{i=1}^{N} d(\boldsymbol{y}^{(i)}, \hat{\boldsymbol{y}}^{(i)}) + \alpha_{sp}\sum_{j=1}^{n_h} d_{sp}(\rho^{(j)}, \bar{\rho}^{(j)}) + c_{reg}\,, \tag{C.1}$$

where $d(\cdot)$ is a specified distance function (identical to that in Eq. (B.1)); $\rho^{(j)}$ is a specified target activation value for each neuron in the hidden layers and $\bar{\rho}^{(j)}$ is the average activation value (averaged over all samples in the training dataset); $d_{sp}(\cdot)$ is a specified distance function (for sigmoid neurons, $d_{sp}(\cdot)$ can be the Kullback–Leibler divergence); $n_h = -n_{k_{cod}} + 2\sum_{j=1}^{k_{cod}} n_j$ is the total number of neurons in the hidden layers; α_{sp} is a parameter controls the influence of the sparsity regularization term; c_{reg} denotes other regularization term that might be used (e.g., L_1 or L_2 regularization).

References

[1] Øksendal B. Stochastic differential equations: an introduction with applications. Berlin: Springer; 2010.
[2] Higham Desmond J. An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Rev 2001;43(3):525–46.
[3] Grigoriu Mircea. Stochastic calculus: applications in science and engineering. Springer; 2005.
[4] Zhang Yuanjin, Kougioumtzoglou Ioannis A, Kong Fan. A Wiener path integral technique for determining the stochastic response of nonlinear oscillators with fractional derivative elements: A constrained variational formulation with free boundaries. Probab Eng Mech 2023;71:103410.
[5] Petromichelakis Ioannis, Kougioumtzoglou Ioannis A. Addressing the curse of dimensionality in stochastic dynamics: A Wiener path integral variational formulation with free boundaries. Proc R Soc A 2020;476(2243):20200385.
[6] Das Sourav, Tesfamariam Solomon. Reliability assessment of stochastic dynamical systems using physics informed neural network based PDEM. Reliab Eng Syst Saf 2024;243:109849.
[7] Kougioumtzoglou Ioannis A, Spanos Pol D. Response and first-passage statistics of nonlinear oscillators via a numerical path integral approach. J Eng Mech 2012;139(9):1207–17.
[8] Bergman LA, Shinozuka M, Bucher CG, Sobczyk K, Dasgupta G, Spanos PD, Deodatis G, Spencer BF, Ghanem RG, Sutoh A, Grigoriu M, Takada T, Hoshiya M, Wedig WV, Johnson EA, Wojtkiewicz SF, Naess A, Yoshida I, Pradlwarter HJ, Zeldin BA, Schuëller GI, Zhang R. A state-of-the-art report on computational stochastic mechanics. Probab Eng Mech 1997;12(4):197–321.
[9] Soong TT, Grigoriu Mircea. Random vibration of mechanical and structural systems. Prentice Hall; 1993.
[10] Roberts JB, Spanos PD. Random vibration and statistical linearization. Dover; 2003.
[11] Li Jie, Chen Jianbing. Stochastic dynamics of structures. Wiley; 2009.
[12] Zhou Tong, Peng Yongbo. Active learning and active subspace enhancement for PDEM-based high-dimensional reliability analysis. Struct Saf 2021;88:102026.
[13] Wang Ziqi. Optimized equivalent linearization for random vibration. Struct Saf 2024;106:102402.
[14] Ditlevsen Ove, Madsen Henrik O. Structural reliability methods, vol. 178, Wiley New York; 1996.
[15] Barlow RE, F. Proschan. Mathematical theory of reliability. Philadelphia: SIAM; 1996.
[16] Freudenthal AM, Garrelts JM, Shinozuka M, Columbia Univ New York Inst for the Study of Fatigue, and Reliability. The analysis of structural safety. Technical report, Institute for the Study of Fatigue and Reliability. Defense Technical Information Center; 1964.
[17] Der Kiureghian Armen. Structural and system reliability. Cambridge University Press; 2022.
[18] Ellingwood Bruce R. Earthquake risk assessment of building structures. Reliab Eng Syst Saf 2001;74(3):251–62.
[19] Frangopol Dan M, Maute Kurt. Life-cycle reliability-based optimization of civil and aerospace structures. Comput Struct 2003;81(7):397–410.
[20] Moller Bernd, Beer Michael. Engineering computation under uncertainty - capabilities of non-traditional models. Comput Struct 2008;86(10):1024–41.
[21] Jerez DJ, Jensen HA, Beer M. Reliability-based design optimization of structural systems under stochastic excitation: an overview. Mech Syst Signal Process 2022;166:108397.
[22] Wang Lei, Liu Yaru, Li Min. Time-dependent reliability-based optimization for structural-topological configuration design under convex-bounded uncertain modeling. Reliab Eng Syst Saf 2022;221:108361.
[23] Kumar Anil, Parkash Chander, Vashishtha Govind, Tang Hesheng, Kundu Pradeep, Xiang Jiawei. State-space modeling and novel entropy-based health indicator for dynamic degradation monitoring of rolling element bearing. Reliab Eng Syst Saf 2022;221:108356.
[24] Jordan Richard, Kinderlehrer David, Otto Felix. The variational formulation of the Fokker–Planck equation. SIAM J Math Anal 1998;29(1):1–17.
[25] Crandall Stephen H. Non-Gaussianclosure techniques for stationary random vibration. Int J Non-Linear Mech 1985;20(1):1–8.
[26] Oden Tinsley, Moser Robert, Ghattas Omar. Computer predictions with quantified uncertainty, part I. SIAM News 2010;43(9):1–3.
[27] Elishakoff I, Sarlin N. Uncertainty quantification based on pillars of experiment, theory, and computation. Part I: Data analysis. Mech Syst Signal Process 2016;74:29–53.
[28] Roy Christopher J, Oberkampf William L. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. Comput Methods Appl Mech Engrg 2011;200(25):2131–44.
[29] Owhadi H, Scovel C, Sullivan T, McKerns M, Ortiz M. Optimal uncertainty quantification. SIAM Rev 2013;55(2):271–345.
[30] Peherstorfer Benjamin, Willcox Karen, Gunzburger Max. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. SIAM Rev 2018;60(3):550–91.

[31] Couaillier Vincent, Savin Éric. Generalized polynomial chaos for non-intrusive uncertainty quantification in computational fluid dynamics. In: Hirsch Charles, Wunsch Dirk, Szumbarski Jacek, Łaniewski-Wołłk Łukasz, Pons-Prats Jordi, editors. Uncertainty management for robust industrial design in aeronautics : findings and best practice collected during UMRIDA, a collaborative research project (2013–2016) funded by the European union. Cham: Springer International Publishing; 2019, p. 123–41.

[32] Eldred M S. Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design. In: 50th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference. 2009.

[33] Yasar Osman, Deng Yuefan, Tuzun R E, Saltz D. New trends in high performance computing. In: IEEE international conference on high performance computing data and analytics. Vol. 27, 2001, p. 3–35.

[34] Shalf John. The future of computing beyond Moore's law. Philos. Trans. R. Soc. A 2020;378(2166):20190061.

[35] Marinescu Dan C. Cloud computing: theory and practice. Morgan Kaufmann; 2022.

[36] Hennigh Oliver, Narasimhan Susheela, Nabian Mohammad Amin, Subramaniam Akshay, Tangsali Kaustubh, Fang Zhiwei, Rietmann Max, Byeon Wonmin, Choudhry Sanjay. NVIDIA SimNet™: An AI-accelerated multi-physics simulation framework. In: International conference on computational science. Springer; 2021, p. 447–61.

[37] Rubinstein RY, Kroese DP. Simulation and the Monte Carlo method. Wiley; 2007.

[38] Neal Radford M. Annealed importance sampling. Statist Comput 2001;11(2):125–39.

[39] Au SK, Ching J, Beck JL. Application of subset simulation methods to reliability benchmark problems. Struct Saf 2007;29(3):183–93.

[40] Wang Ziqi, Broccardo Marco, Song Junho. Hamiltonian Monte Carlo methods for subset simulation in reliability analysis. Struct Saf 2019;76:51–67.

[41] Xian Jianhua, Wang Ziqi. Relaxation-based importance sampling for structural reliability analysis. Struct Saf 2024;106:102393.

[42] Wang Ziqi, Song Junho. Cross-entropy-based adaptive importance sampling using von Mises–Fisher mixture for high dimensional reliability analysis. Struct Saf 2016;59:42–52.

[43] Papaioannou Iason, Geyer Sebastian, Straub Daniel. Improved cross entropy-based importance sampling with a flexible mixture model. Reliab Eng Syst Saf 2019;191:106564.

[44] Betz Wolfgang, Papaioannou Iason, Straub Daniel. Bayesian post-processing of Monte Carlo simulation in reliability analysis. Reliab Eng Syst Saf 2022;227:108731.

[45] Xiu Dongbin. Numerical methods for stochastic computations: A spectral method approach. Princeton University Press; 2010.

[46] Echard B, Gayton N, Lemaire M. AK-MCS: An active learning reliability method combining Kriging and Monte Carlo simulation. Struct Saf 2011;33(2):145–54.

[47] Sudret Bruno. Meta-models for structural reliability and uncertainty quantification. 2012, arXiv:1203.2062.

[48] Xu Yanwen, Kohtz Sara, Boakye Jessica, Gardoni Paolo, Wang Pingfeng. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. Reliab Eng Syst Saf 2023;230:108900.

[49] Xu Zhaoyi, Saleh Joseph Homer. Machine learning for reliability engineering and safety applications: Review of current status and future opportunities. Reliab Eng Syst Saf 2021;211:107530.

[50] Afshari Sajad Saraygord, Enayatollahi Fatemeh, Xu Xiangyang, Liang Xihui. Machine learning-based methods in structural reliability analysis: A review. Reliab Eng Syst Saf 2022;219:108223.

[51] Kim Jungho, Wang Ziqi, Song Junho. Adaptive active subspace-based metamodeling for high-dimensional reliability analysis. Struct Saf 2024;106:102404.

[52] Xian Jianhua, Wang Ziqi. A physics and data co-driven surrogate modeling method for high-dimensional rare event simulation. 2023, arXiv preprint arXiv:2310.00261.

[53] Chakroborty Promit, Dhulipala Somayajulu LN, Che Yifeng, Jiang Wen, Spencer Benjamin W, Hales Jason D, Shields Michael D. General multifidelity surrogate models: Framework and active-learning strategies for efficient rare event simulation. J Eng Mech 2023;149(12):04023096.

[54] Crauel Hans, Debussche Arnaud, Flandoli Franco. Random attractors. J Dyn Differ Equ 1997;9(2):307–41.

[55] Roberts AJ. The utility of an invariant manifold description of the evolution of a dynamical system. SIAM J Math Anal 2006;20(6):1447–58.

[56] Song Junho, Kiureghian Armen Der. Bounds on system reliability by linear programming. J Eng Mech 2003;129(6):627–36.

[57] Song Junho, Kang Won Hee. System reliability and sensitivity under statistical dependence by matrix-based system reliability method. Struct Saf 2009;31(2):148–56.

[58] Koo Heonsang, Der Kiureghian Armen, Fujimura Kazuya. Design-point excitation for non-linear random vibrations. Probab Eng Mech 2005;20(2):136–47.

[59] Fujimura Kazuya, Der Kiureghian Armen. Tail-equivalent linearization method for nonlinear random vibration. Probab Eng Mech 2007;22(1):63–76.

[60] Broccardo Marco. Further development of the tail-equivalent linearization method for nonlinear stochastic dynamics (Ph.D. thesis), UC Berkeley; 2014.

[61] Garrè Luca, Der Kiureghian Armen. Tail-equivalent linearization method in frequency domain and application to marine structures. Mar Struct 2010;23(3):322–38.

[62] Wang Ziqi, Der Kiureghian Armen. Tail-equivalent linearization of inelastic multisupport structures subjected to spatially varying stochastic ground motion. J Eng Mech 2016;142(8):04016053.

[63] Alibrandi Umberto, Mosalam Khalid M. Equivalent linearization methods for stochastic dynamic analysis using linear response surfaces. J Eng Mech 2017;143(8):04017055.

[64] Broccardo Marco, Der Kiureghian Armen. Nonlinear stochastic dynamic analysis by evolutionary tail-equivalent linearization method. Struct Saf 2021;90:102044.

[65] Ma Yunqian, Fu Yun. Manifold learning theory and applications, vol. 434, CRC press Boca Raton; 2012.

[66] Soize Christian, Ghanem Roger. Data-driven probability concentration and sampling on manifold. J Comput Phys 2016;321:242–58.

[67] Soize Christian, Ghanem Roger. Physics-constrained non-Gaussian probabilistic learning on manifolds. Internat J Numer Methods Engrg 2020;121(1):110–45.

[68] Soize Christian, Ghanem Roger. Probabilistic learning on manifolds constrained by nonlinear partial differential equations for small datasets. Comput Methods Appl Mech Engrg 2021;380:113777.

[69] Soize Christian, Ghanem Roger. Probabilistic learning on manifolds (PLoM) with partition. Internat J Numer Methods Engrg 2022;123(1):268–90.

[70] Giovanis Dimitris G, Shields Michael D. Uncertainty quantification for complex systems with very high dimensional response using grassmann manifold variations. J Comput Phys 2018;364:393–415.

[71] Giovanis Dimitris G, Shields Michael D. Data-driven surrogates for high dimensional models using Gaussian process regression on the Grassmann manifold. Comput Methods Appl Mech Engrg 2020;370:113269.

[72] Kontolati Katiana, Loukrezis Dimitrios, dos Santos Ketson RM, Giovanis Dimitrios G, Shields Michael D. Manifold learning-based polynomial chaos expansions for high-dimensional surrogate models. Int J Uncertain Quantif 2022;12(4).

[73] Dos Santos Ketson R, Giovanis Dimitrios G, Shields Michael D. Grassmannian diffusion maps–based dimension reduction and classification for high-dimensional data. SIAM J Sci Comput 2022;44(2):B250–74.

[74] Lataniotis Christos, Marelli Stefano, Sudret Bruno. Extending classical surrogate modeling to high dimensions through supervised dimensionality reduction: a data-driven approach. Int J Uncertain Quantif 2020;10(1).

[75] Schär Styfen, Marelli Stefano, Sudret Bruno. Emulating the dynamics of complex systems using autoregressive models on manifolds (mNARX). Mech Syst Signal Process 2024;208:110956.

[76] Au SK, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2001;16(4):263–77.

[77] Cérou Frédéric, Del Moral Pierre, Furon Teddy, Guyader Arnaud. Sequential Monte Carlo for rare event estimation. Stat Comput 2012;22(3):795–808.

[78] Moral Pierre Del, Doucet Arnaud, Jasra Ajay. Sequential Monte Carlo samplers. J R Statist Soc Ser B 2006;68(3):411–36.

[79] Chen Weiming, Wang Ziqi, Broccardo Marco, Song Junho. Riemannian manifold Hamiltonian Monte Carlo based subset simulation for reliability analysis in non-Gaussian space. Struct Saf 2022;94:102134.

[80] Van Der Maaten Laurens, Postma Eric O, van den Herik H Jaap, et al. Dimensionality reduction: A comparative review. J Mach Learn Res 2009;10(66–71):13.

[81] Zebari Rizgar, Abdulazeez Adnan, Zeebaree Diyar, Zebari Dilovan, Saeed Jwan. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. J Appl Sci Technol Trends 2020;1(2):56–70.

[82] Coifman Ronald R, Lafon Stephane, Lee Ann B, Maggioni Mauro, Nadler Boaz, Warner Frederick, Zucker Steven W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proc Natl Acad Sci USA 2005;102(21):7426–31.

[83] Coifman Ronald R, Lafon Stéphane. Diffusion maps. Appl Comput Harmonic Anal 2006;21(1):5–30.

[84] Hinton Geoffrey E, Salakhutdinov Ruslan. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7.

[85] Vincent Pascal, Larochelle Hugo, Lajoie Isabelle, Bengio Yoshua, Manzagol Pierreantoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 2010;11:3371–408.

[86] Everitt BSBS, Landau S, Leese M. Cluster analysis. John Wiley & Sons; 2011.

[87] Rousseeuw Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20(1):53–65.

[88] Sugar Catherine A, James Gareth M. Finding the number of clusters in a dataset: An information-theoretic approach. J Amer Statist Assoc 2003;98(463):750–63.

[89] Amorim Renato Cordeiro De, Hennig Christian. Recovering the number of clusters in data sets with noise features using feature rescaling factors. Inform Sci 2015;324:126–45.

[90] Dempster Arthur P, Laird Nan M, Rubin Donald B. Maximum likelihood from incomplete data via the EM algorithm. J R Statist Soc Ser B 1977;39(1):1–22.

[91] Moller Martin Fodslette. Original contribution: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 1993;6(4):525–33.

[92] Lorenz Edward N. Deterministic nonperiodic flow. J Atmos Sci 1963;20(2):130–41.
[93] Matson Leslie E. The Malkus–Lorenz water wheel revisited. Am J Phys 2007;75(12):1114–22.
[94] Campello Ricardo J G B, Moulavi Davoud, Zimek Arthur, Sander Jorg. Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Trans Knowl Discov Data 2015;10(1):5.
[95] Clough RW, Penzien J. Dynamics of structures. Machw Hill; 1975, p. 634.
[96] C Chatfield. The analysis of time series: An introduction. (6th ed.). CRC Press LLC; 2004.
[97] Wang Ziqi, Broccardo Marco, Song Junho. Hamiltonian Monte Carlo methods for subset simulation in reliability analysis. Struct Saf 2019;76:51–67.
[98] Neal Radford M, et al. MCMC using Hamiltonian dynamics. Handb Markov Chain Monte Carlo 2011;2:113–62.
[99] Hall Peter, Park Byeong U, Samworth Richard J. Choice of neighbor order in nearest-neighbor classification. Ann Statist 2008;36(5):2135–52.
[100] Olshausen Bruno A, Field David J. Sparse coding with an overcomplete basis set: A strategy employed by V1? Vis Res 1997;37(23):3311–25.