14th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP14
Dublin, Ireland, July 9-13, 2023

Stacked Sparse Autoencoders and Classical Artificial Neural Networks for the Inverse Quantification of Mixed Aleatory and Epistemic Uncertainties in the Mathematical Models of Dynamic Engineering Systems, in the Presence of Functional Data

Nicola Pedroni
Associate Professor, Dept. of Energy, Politecnico di Torino, Torino, Italy

ABSTRACT: The paper proposes an integrated framework for Inverse Uncertainty Quantification (IUQ), based on the effective combination of: Stacked Sparse Autoencoders for the dimensionality reduction of time dependent, scarce data; Artificial Neural Networks metamodels, to reduce the computational burden typically associated to system simulation codes; Genetic Algorithms to solve the inverse optimization problems related to the IUQ process; and Sliced Normal distributions to describe complex, asymmetric and multi-modal dependencies. The approach is shown to perform satisfactorily on the 2020 NASA UQ Challenge: accurate and robust characterizations of uncertainty (together with model discrepancies) are produced, while reducing the overall computational cost by two orders of magnitude.

1. INTRODUCTION

In the analysis of *complex*, *safety-critical* (e.g., civil, nuclear, aerospace and chemical) *dynamic* engineering systems, it is necessary to provide the uncertainty quantification of the outputs of the mathematical models (and the corresponding computer codes) used to simulate the system. Such uncertainty characterization is of paramount importance for: (i) making robust decisions; (ii) optimally designing and operating such systems; and (iii) driving resource allocation for uncertainty reduction. To this aim, the uncertainty affecting the selected inputs needs to be *quantified* (coherently with the information available) and *propagated* through the code (Bi et al., 2022).

The aim of the paper is to propose an integrated framework for the *inverse* (joint) quantification of *heterogeneous* (aleatory and epistemic) input uncertainties by means of code simulation results and raw, experimental output data from the real engineering system under analysis. Emphasis is given to those challenging situations where: (i) the models are *computationally demanding*; (ii) input and/or output variables are *functions* of time and/or space; (iii) the data employed are *scarce* and exhibit complex, strongly nonlinear *dependence* patterns (Wu et al., 2021).

The approach is based on an efficient combination of: (i) Stacked Sparse Autoencoders (SSAEs) to *reduce* the problem *dimensionality* by projecting the time- or space-dependent (output) variables onto a proper feature space (Zhao et al., 2019); (ii) classical Artificial Neural Networks (ANNs) (Dong et al., 2023) to *lower* the *computational burden* and replace the original long-running simulation code by learning the relationship between the model inputs and the SSAE-based projected features (Nanty et al., 2017); (iii) heuristic, global optimization tools (in particular, Genetic Algorithms-GAs) to perform the *inverse identification* (retrieval) of the uncertain input values corresponding to the available raw output data; (iv) Sliced Normal (SN) distributions to fit the retrieved data and finally quantify the overall input uncertainty (Crespo et al., 2019).

The effectiveness and criticalities of the approach are tested on the quantification of mixed aleatory (*probabilistic*) and epistemic (*set-based*) uncertainties affecting the mathematical model of a dynamic aerospace system, proposed within the NASA Langley Uncertainty Quantification

Challenge on Optimization Under Uncertainty (Crespo and Kenny, 2021).

2. THE PROBLEM

Consider a system modelled by the function $y(\boldsymbol{a}, \boldsymbol{e}, t)$, where $\boldsymbol{a}$ is a n_a-dimensional vector of aleatory variables, $\boldsymbol{e}$ is a n_e-dimensional vector of epistemic parameters and t is time. The Uncertainty Model (UM) for $\boldsymbol{a}$ is denoted as $f_{\boldsymbol{a}}$, where $f_{\boldsymbol{a}}$ is a joint density (of *unknown* functional form) supported in the set A. In contrast, the UM for $\boldsymbol{e}$ is denoted as E, where E is a hyper-rectangular set. Hence, the UM is fully prescribed by the pair $\langle f_{\boldsymbol{a}}, E \rangle$. The function y is possibly given as a discrete time history, e.g., $y(t) = [y(0), y(dt), \ldots, y(N_T \cdot dt)]$, where N_T is the number of discrete time steps and $N_T \cdot dt = T$ is the time horizon of the analysis. The objective is to perform an efficient and accurate joint calibration of aleatory and epistemic uncertainty (i.e., to prescribe the UM $\langle f_{\boldsymbol{a}}, E \rangle$), in the presence of a (possibly small-sized) dataset $\boldsymbol{D} = \{y_i(t)\}$, $i = 1, 2, \ldots, N_D$, $t = 0, 1, \ldots, N_T$, and a computationally demanding model y.

3. THE CALIBRATION METHOD

The following steps are performed to prescribe the joint UM $\langle f_{\boldsymbol{a}}, E \rangle$:

1. Dimensionality reduction. Transform the dataset $\boldsymbol{D} = \{y_i(t)\}, t = 0, 1, \ldots, N_T$, into a lower dimensional representation, say $\boldsymbol{s}_i = \{s_{i1} \ldots, s_{ik}, \ldots, s_{iNs}\}$, $i = 1, 2, \ldots, N_D$, using a properly selected function $\boldsymbol{S}(\cdot)$: $R^{N_T+1} \rightarrow R^{Ns}$ ($N_s << N_T$), such that $\boldsymbol{s}_i = \boldsymbol{S}(y_i(t))$. In this paper, Stacked Sparse AutoEncoders (SSAE) (Zhao et al., 2019) are employed due to their proven ability to deal with complex, nonlinear (and possibly noisy) data (see Section 3.1). The idea underlying this pre-processing step is to perform the subsequent calibration task in the (static multivariate) "reduced" space (defined by the transformation function $\boldsymbol{S}$) rather than in the (dynamic multivariate) time domain.
2. Meta-modeling in the reduced space. Construct a (fast-running) metamodel-based approximation to the (long-running) system code implementing the function $y(\boldsymbol{a}, \boldsymbol{e}, t)$, in order to reduce the computational burden associated to the calibration task. The metamodel is built based on a finite (training) set D_{TR} of N_{TR} data representing examples of the input/output nonlinear relationships underlying the original system model code. The generation of this data set D_{TR} entails running the original mathematical model $y(\boldsymbol{a}, \boldsymbol{e}, t)$ a predetermined number of times N_{TR} for specified values $\{(\boldsymbol{a}_j, \boldsymbol{e}_j), j = 1, 2, \ldots, N_{TR}\}$ of the input variables and collecting the corresponding output time series $\{y(\boldsymbol{a}_j, \boldsymbol{e}_j, t)\}$, which are then projected onto the reduced space to get the features $\boldsymbol{s}_j = \{\boldsymbol{s}(\boldsymbol{a}_j, \boldsymbol{e}_j)\}$. Then, statistical techniques (for example, regression error minimization procedures) are employed for adjusting the internal parameters of the regression model to fit the input/output data D_{TR} thereby generated and to capture the underlying (possibly nonlinear and non-monotonic) relationship. Notice that in this paper, *one* dedicated metamodel MM_k is trained to emulate *each* feature s_k, $k = 1, 2, \ldots, N_s$: in other words, a direct mapping is established between the inputs and the outputs in the *reduced* space, rather than in the time domain (Nanty et al., 2017). Once built, the surrogate model can be used for performing, in an acceptable computational time, the numerous repeated evaluations needed for an accurate model calibration and uncertainty quantification. In this work, ANNs regression models (Dong et al., 2023) are used to this purpose. From a mathematical viewpoint, ANNs consist of a set of nonlinear (e.g., sigmoidal) basis functions with adaptable parameters that are adjusted by the process of training. ANNs have been demonstrated to be universal approximants of continuous nonlinear functions (under mild mathematical conditions), i.e., in principle, an ANN model with a properly selected architecture can be a consistent estimator of any continuous nonlinear function. Further details about ANN regression models are not reported here for

brevity; the interested reader may refer to the cited references and the literature in the field.

3. Quantification of the uncertainty in the epistemic parameters. In the reduced space, fit a joint PDF $f_S(\cdot)$ to the N_D available time series observations $\{y_i(t)\}$ under the transformation $\boldsymbol{S}(y_i(t))$: at this stage, a simple multivariate Kernel Density Estimator (KDE) (e.g., the product Gaussian kernel) can be employed to capture the main characteristics and dependences of the dataset. Using the distribution $f_S(\cdot)$, the likelihood $f_S(\boldsymbol{S}(y(\boldsymbol{a}, \boldsymbol{e}, t)))$ can be assessed for any $(\boldsymbol{a}, \boldsymbol{e}) \in A \times E$. For a point $\boldsymbol{e} \in E$ to be *plausible*, it should be possible to find some $\boldsymbol{a} \in A$ such that $f_S(\boldsymbol{S}(y(\boldsymbol{a}, \boldsymbol{e}, t)))$ is relatively large (or at least *not negligible*). That is, there must exist some $\boldsymbol{a} \in A$ such that $\boldsymbol{S}(y(\boldsymbol{a}, \boldsymbol{e}, t))$ lies within the main bulk of the distribution $f_S(\cdot)$. In fact, for each observation $\{y_i(t)\}$, there must exist *some* $\boldsymbol{a}_i \in A$ such that $\boldsymbol{S}(y(\boldsymbol{a}_i, \boldsymbol{e}, t))$ corresponds with $\boldsymbol{S}(y_i(t))$, in order for $\boldsymbol{e}$ to be *plausible*. In practice, a set of epistemic values $\boldsymbol{e}_l$, $l = 1, 2, \ldots, N_e$, is selected (either deterministically or stochastically) in E, and a relatively large sequence of aleatory values $\boldsymbol{a}_j$, $j = 1, 2, \ldots, N_a$, is uniformly sampled to thoroughly probe A (as described in Section 2, *no information* at all is available on $f_{\boldsymbol{a}}$: correspondingly, no hypothesis is made in this respect by the analyst). Using the joint PDF $f_S(\cdot)$ fitted to the available data in the reduced space $\boldsymbol{S}(y_i(t))$, the plausibility $\mathcal{L}(\boldsymbol{e}_l)$ of each $\boldsymbol{e}_l$ is computed as its likelihood $1/N_a \cdot \sum_{j=1}^{N_a} f_{\boldsymbol{S}}\left(\boldsymbol{S}(\boldsymbol{a}_j, \boldsymbol{e}_k, t)\right)$ "averaged" over the aleatory space. Finally, the function $\mathcal{L}(\boldsymbol{e})$ is approximated over E and normalized to a PDF; then, the epistemic UM E is defined as the smallest hyper-rectangle enveloping the joint n_e-dimensional α% Confidence Interval (CI) of $\boldsymbol{e}$. Notice that the confidence level α can be selected by the analyst as a trade-off between model robustness (conservatism) and predictivity.
4. Quantification of the uncertainty in the aleatory variables. The approach here proposed is based on the (repeated) calculation of several solutions to the *inverse optimization* problem of identifying (*retrieving*) the aleatory input values $\boldsymbol{a}_i$ that presumably "generated" the available raw output data $\{y_i(t)\}$, projected onto the reduced feature space $\boldsymbol{s}_i = \boldsymbol{S}(y_i(t))$. In detail, select the epistemic value of $\boldsymbol{e}$ characterized by the maximum plausibility $\boldsymbol{e}_{opt}$. Then, for each datum $\boldsymbol{s}_i = \boldsymbol{S}(y_i(t))$, $i = 1, 2, \ldots, N_D$, solve the following minimization problem:

$$\boldsymbol{a}_i^{opt} = \arg\min_{\boldsymbol{a}_i \in A} \left\{ \left\| \boldsymbol{s}_i - \boldsymbol{S}\left(y(\boldsymbol{a}_i, \boldsymbol{e}_{opt}, t)\right) \right\| \right\} \quad (1)$$

where $\|\cdot\|$ is the L^p norm (here, $p = 2$). Hence, $\boldsymbol{a}_i^{opt}$ is the aleatory vector that tries to attain a zero-prediction error for the i-th (projected) datum in the sequence, when the epistemic parameter is $\boldsymbol{e}_{opt}$. Genetic Algorithms (GAs) (that are heuristic, global optimization tools) are employed in this paper to solve (1). The solutions to the N_D problems (1) represent a sequence of N_D aleatory vectors $\{\boldsymbol{a}_i: i = 1, 2, \ldots, N_D\}^{opt}$ that can be used to construct (i.e., to fit) *a posteriori* the joint multivariate PDF $f_{\boldsymbol{a}}(\boldsymbol{a}|\boldsymbol{e}_{opt})$. In the present paper, Sliced Normal (SN) distributions (Crespo et al., 2019) are employed, due to their proven ability and versatility in characterizing complex dependencies with reduced modeling effort (Section 3.2). Finally, to build robustness in the characterization of aleatory uncertainty, additional epistemic values $\boldsymbol{e}_k$, $k = 1, 2, \ldots, N_e$, can be selected in the refined hyper-rectangle E (defined at step 3. above); for each $\boldsymbol{e}_k$ the optimization problems (1) are repeatedly solved to obtain N_e sets of aleatory vectors $\{\boldsymbol{a}_i: i = 1, 2, \ldots, N_D\}^k$, $k = 1, 2, \ldots, N_e$. *Each* set is used to construct a PDF $f_{\boldsymbol{a}}(\boldsymbol{a}|\boldsymbol{e}_k)$, which represents an aleatory model compatible with the available data/information and of plausibility $\mathcal{L}(\boldsymbol{e}_k)$ defined by the selected $\boldsymbol{e}_k$.
5. Quantification of model uncertainty (or discrepancy). Often, the calibrated model provides an insufficient fit to the experimental data. Model error is unavoidable in many situations due to an incomplete understanding of underlying physics, likely in addition to

large and possibly poorly characterized uncertainties in the calibration data. This type of *discrepancy* is generally attributed to model form error or structural error. Such models can then be corrected to some extent by including a model discrepancy term (Maupin and Swiler, 2020). An additive discrepancy formulation is often introduced, here extended to account for the time dependent nature of the outputs, i.e., $y_i(t) = y(\boldsymbol{a}_i, \boldsymbol{e}, t) + \delta(\boldsymbol{a}_i, t) + \varepsilon_i(t)$, where $\varepsilon_i(t)$ is the experimental noise and $\delta(\boldsymbol{a}_i, t)$ is the discrepancy (or model uncertainty) term. For convenience, in this paper the discrepancy term is originally quantified in the space of reduced dimensionality (Step 1 above), i.e., $\boldsymbol{s}_i = \boldsymbol{S}(y_i(t)) = \boldsymbol{S}(y(\boldsymbol{a}_i, \boldsymbol{e}, t)) + \boldsymbol{\delta}_s(\boldsymbol{a}_i) + \boldsymbol{\varepsilon}_{si}(t)$, $i = 1, 2, \ldots, N_D$. Exploiting the results obtained at Step 4 by the optimization-based input retrieval, *explicit* values for $\boldsymbol{\delta}_s$ can be computed as $\boldsymbol{\delta}_s(\boldsymbol{a}_i|\boldsymbol{e}_k) = \boldsymbol{s}_i - \boldsymbol{S}(y(\boldsymbol{a}_i, \boldsymbol{e}_k, t))$, $i = 1, 2, \ldots, N_D$, (in principle) for all the different aleatory models k obtained in correspondence of the different epistemic vectors $\boldsymbol{e}_k$, $k = 1, 2, \ldots, N_e$. It is assumed that the experimental noise has zero mean. The calculated values of $\boldsymbol{\delta}_s(\boldsymbol{a}_i|\boldsymbol{e}_k)$ are therefore used to calibrate discrepancy model(s) $\widehat{\boldsymbol{\delta}}_{\boldsymbol{s}}(\boldsymbol{a}|\boldsymbol{e}_k)$ in the reduced space. The functional form that is most appropriate for $\boldsymbol{\delta}_s$ obviously depends on the application at hand: in facts, there is a general lack of consensus about a coherent mathematical formulation of model uncertainty (discrepancy). Gaussian processes are often adopted in the literature; however, in this work, the flexibility and modeling power of ANN metamodels are used to *avoid* relying on the *Gaussian hypothesis*. Also, uncertainty in the discrepancy predictions can be assessed by bootstrapping or the delta method.

6. Final UM. Deliver the final UM as $\langle f_{\boldsymbol{a}}(\boldsymbol{a}|\boldsymbol{e}_k), E\rangle$ with the corresponding discrepancy functions estimated in the reduced space $\widehat{\boldsymbol{\delta}}_{\boldsymbol{s}}(\boldsymbol{a}|\boldsymbol{e}_k)$, for $k = 1, 2, \ldots, N_e$, including $\boldsymbol{e} = \boldsymbol{e}_{opt}$.

Notice that the quantification of aleatory uncertainty presented at Step 4 has the following advantages: (i) it does *not* require any limiting assumption or hypothesis to define *a priori* a given functional form for $f_{\boldsymbol{a}}$, but rather it optimally fits *a posteriori* the pointwise input configurations retrieved by (1); (ii) it does *not* require the cumbersome *repeated propagation* of input uncertainties (necessary to the evaluation of the likelihood), which instead characterizes classical Bayesian inversion (in particular, when both aleatory and epistemic uncertainties are modeled within a non-parametric framework); (iii) since it based on the solution of N_D optimization problems, it is *computationally affordable* in the presence of *scarce data*, which is of paramount interest to the present paper; (iv) the quality of the optimization results (i.e., the prediction errors attained during the inputs retrieval) provides a "visual", quantitative evidence of the *necessity* (or not) of estimating *model errors* (or *discrepancies*); (v) in such a case, the explicit retrieval of the input vectors $\boldsymbol{a}_i$ allows a *direct fit* of the *discrepancy* $\boldsymbol{\delta}_s(\boldsymbol{a}_i)$, without subjective a priori assumptions about its functional form.

3.1. Stacked Sparse AutoEncoders (SSAEs)

An autoencoder (AE) is a type of Artificial Neural Network (ANN) that is designed to learn new *low-dimensional* latent features of the data by trying to reconstruct the input data. It is composed of an *encoder* network, that maps an N_T-dimensional vector (time series) $y(t)$ into a lower-dimensional (namely, hidden) space of N_1 features $\boldsymbol{s}_1$ ($N_1 << N_T$); and a *decoder* network that transforms the hidden representation $\boldsymbol{s}_1$ back into $\hat{y}(t)$ (i.e, the reconstruction of $y(t)$) (Zhao et al., 2019). The Sparse AE (SAE), which is a variation of the AE, imposes sparsity constraints on the hidden layer of the network to encourage the identification of only discriminative features. The determination of the SAE internal regression parameters is carried out through the minimization of the following cost function: $E = R_{error} + \beta R_{sparse} + \lambda R_{L2}$, where R_{error}, R_{sparse} and R_{L2} are the reconstruction error, the sparsity and the *L2* regularization terms, respectively, while β and λ are coefficients indicating the relative importance of the terms in the cost function. Specifically, the R_{error} quantifies

the dissimilarity between the input vector and its reconstruction; the R_{sparse} measures the average "activation" of the hidden neurons (actually, it has been shown that the extraction of discriminative hidden features $\boldsymbol{s}_1$ is favored by requiring the sparsity of the AE, which imposes the neurons to be mostly "inactive", i.e., to be characterized by relatively low output values); R_{L2} is introduced to prevent SAE overfitting.

To improve the representational and modeling power, multiple pretrained AEs can be stacked to form a multiple-hidden-layer ANN, called Stacked Sparse Autoencoder (SSAE). Since training non-linear autoencoders with multiple hidden layers is a difficult task, the following breakthrough approach is adopted, which consists of a *pre-training* and a *fine-tuning* phase: (1) The pre-training regards a consecutive training of S (basic) SAEs. Initially, the first basic SAE is trained using the time series $y_i(t)$; then, the corresponding extracted features $\boldsymbol{s}_{i,1}$ (of dimension N_1) are used as training input vectors for the next (second) basic SAE, which transforms $\boldsymbol{s}_{i,1}$ into $\boldsymbol{s}_{i,2}$ (of dimension $N_2 < N_1$). This step is repeated up to the last (S-th) SAE, which is trained to deliver the hidden features $\boldsymbol{s}_{i,S}$ (of dimension N_s). Then, the SSAE is built by stacking all the basic SAEs. (2) In the fine-tuning phase, the SSAE obtained from the pre-training is fine-tuned using the classical ANN backpropagation of error derivatives. The *encoder* part of the SSAE thereby trained is used to carry out the dimensionality reduction from $y_i(t)$ to the features $\boldsymbol{s}_{iS}$ of dimension N_s (referred to as $\boldsymbol{s}_i$ for the sake of compact notation), to be used in the calibration.

3.2. Sliced Normal (SN) distributions

Sliced Normal (SN) distributions are a generalization of Gaussian distributions where the quadratic argument of the exponential is replaced with a sum of squares polynomial. In particular:

$$f(\boldsymbol{a};\boldsymbol{\mu},\boldsymbol{P},d) \sim \frac{exp(-1/2\phi(\boldsymbol{a};\boldsymbol{\mu},\boldsymbol{P},d))}{(2\pi)^{n_w/2}\sqrt{|\boldsymbol{P}^{-1}|}} \quad (2)$$

where $\boldsymbol{\mu}$ is the mean vector (of size n_w); $\boldsymbol{P}$ is a positive semi-definite matrix (of size n_w x n_w); $\phi(\boldsymbol{a};\boldsymbol{\mu},\boldsymbol{P},d) = (\boldsymbol{W}_d(\boldsymbol{a})-\boldsymbol{\mu})^T \cdot \boldsymbol{P} \cdot (\boldsymbol{W}_d(\boldsymbol{a})-\boldsymbol{\mu})$ with $\boldsymbol{W}_d(\boldsymbol{a})$ a vector of monomials in variables $\boldsymbol{a}$ of degree greater than zero and less than or equal to d (to be defined by the analyst). The monomials of $\boldsymbol{W}_d(\boldsymbol{a})$ are in graded lexicographic order: they are first ordered by the canonical order in the degree, and second by using lexicographic order. For example, if $n_a = 2$ and $d = 2$, $\boldsymbol{W}_2(\boldsymbol{a}) = [a_1, a_2, a_1^2, a_1 \cdot a_2, a_2^2]^T$. In other words, $\boldsymbol{W}_d(\boldsymbol{a})$ performs a polynomial mapping from the physical space $\boldsymbol{a}$ (of dimension n_a) to a "lifted" feature space $\boldsymbol{w}$ (of dimension $n_w = \binom{n_a+d}{n_a} - 1$). In this way, variables $\boldsymbol{a}$ do *not* have a Gaussian distribution, while only the transformed variables $\boldsymbol{w}$ do. Naturally, when $d = 1$, (2) reduces to the classical Normal distribution. Instead, when $d > 1$, SNs are shown capable of representing a diverse set of random variables including multi-modal, non-symmetric, and skewed distributions. Optimal values $\boldsymbol{\mu}*$ and $\boldsymbol{P}*$ of the adaptable parameters $\boldsymbol{\mu}$ and $\boldsymbol{P}$ can be easily calculated by Maximum Likelihood Estimation (MLE) in the feature space $\boldsymbol{w}$ and are demonstrated to have explicit analytical forms: see (Crespo et al., 2019) for technical details.

The optimal PDF (2) can be employed to obtain semi-algebraic representations of nested, closed confidence regions of the form $S(\beta_\alpha) = \{\boldsymbol{a}: (\boldsymbol{W}_d(\boldsymbol{a})-\boldsymbol{\mu})T \cdot \boldsymbol{P} \cdot (\boldsymbol{W}_d(\boldsymbol{a})-\boldsymbol{\mu}) \leq \beta_\alpha\}$, where, for a desired confidence level α, β_α must be numerically calculated such that $\alpha \cdot 100\%$ of the data is contained in $S(\beta_\alpha)$.

4. CASE STUDY

The proposed method is here applied to solve the 2020 NASA Langley challenge on optimization under uncertainty (subproblem A), aimed at characterizing the uncertain response of a system controlled by a non-collocated sensor/actuator (Crespo and Kenny, 2021). The system of interest is modelled as a set of interconnected subsystems. The first subsystem is modelled by the black-box function $y(\boldsymbol{a}, \boldsymbol{e}, t)$, where $\boldsymbol{a}$ is an n_a-dimensional vector of aleatory variables ($n_a = 5$), $\boldsymbol{e}$ is a n_e-dimensional vector of epistemic parameters ($n_e = 4$). The UM for $\boldsymbol{a}$ is denoted as $f_{\boldsymbol{a}}$, where $f_{\boldsymbol{a}}$ is a joint density supported in the set $A_0 = [0, 2]^{na}$. In contrast, the UM for $\boldsymbol{e}$ is denoted as E, where E is

a hyper-rectangular set included in $E_0 = [0, 2]^{ne}$. The integrated system is instead modeled by $\boldsymbol{z}(\boldsymbol{a}, \boldsymbol{e}, t) = \{z_1(\boldsymbol{a}, \boldsymbol{e}, t), z_2(\boldsymbol{a}, \boldsymbol{e}, t)\}$. Hence, the output of the subsystem is a function of time, whereas the output of the integrated system are two functions of time. These functions are given as discrete time histories, e.g., $y(t) = [y(0), y(dt), \ldots, y(N_T \cdot dt)]$, with $N_T = 5000$ and $N_T \cdot dt = T = 5$ s. The objective is to prescribe $\langle f_{\boldsymbol{a}}, E \rangle$ using two datasets: $\boldsymbol{D}(y) = \{y_i(t): i = 1, 2, \ldots, N_D(y) = 100\}$ from the subsystem, and $\boldsymbol{D}(z_1, z_2) = \{(z_{1,i}(t), z_{2,i}(t)): i = 1, 2, \ldots, N_D(z_1, z_2) = 100\}$ from the integrated system.

5. RESULTS

The results of all the different steps of the IUQ method (Section 3) are reported hereafter.

Three SSAEs are trained to reproduce (and reduce the dimensionality of) the three sets of $N_D(y)$, $N_D(z_1)$ and $N_D(z_2)$ time series, which results in $N_s(y) = 10$, $N_s(z_1) = 13$ and $N_s(z_2) = 4$ projected features $\{\boldsymbol{s}_i^y\}$, $i = 1, 2, \ldots, N_D(y)$, $\{\boldsymbol{s}_j^{z_1}\}$ and $\{\boldsymbol{s}_j^{z_2}\}$, $j = 1, 2, \ldots, N_D(z_1) = N_D(z_2) = 100$, respectively (Step 1). The SSAEs architectures and hyperparameters (optimized by trial-and-error) are reported in Table 1, together with the percentage Normalized Root Mean Square Errors (NRMSEs) of reconstruction, R_{error}.

Table 1. SSAEs training results

	$\boldsymbol{y}$	$\boldsymbol{z_1}$	$\boldsymbol{z_2}$
S	3	3	3
N_1	500	500	500
N_2	50	50	50
$N_s = N_3$	10	13	4
ρ	0.5	0.3	0.2
β	0.4	0.4	0.08
λ	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$1 \cdot 10^{-4}$
R_{error}	1.52%	0.62%	0.85%

The ANN surrogate models for $y(\cdot)$, $z_1(\cdot)$ and $z_2(\cdot)$ in the SSAE feature space are built based on three training, validation and test sets D_{TR}, D_{VAL} and D_{TEST} of sizes $N_{TR} = 70000$, $N_{VAL} = 20000$ and $N_{TEST} = 10000$, respectively, generated by Latin Hypercube Sampling-LHS (Step 2). The architectures (i.e., the number of hidden neurons) of $N_s(y) + N_s(z_1) + N_s(z_2) = 10 + 13 + 4 = 27$ four-layered feed-forward ANNs (i.e., one for each feature) are selected as those minimizing the NRMSEs on the respective validation sets D_{VAL}'s; finally, the ANNs thereby trained produce NRMSE values smaller than 0.8% for all the features on the corresponding test sets D_{TEST}'s. The time needed to train the ANNs is approximately 10 hours on an Intel(R) Xeon(R) E5-2637 v3 CPU@3.50GHz. The use of ANNs in the calibration process (instead of the original models) leads to a reduction in the computational cost by two orders of magnitude.

The product Gaussian kernel is used to fit two joint PDFs $f_{\boldsymbol{s}^y}(\cdot)$ and $f_{\boldsymbol{s}^z}(\cdot)$ to the $N_D(y) = 100$ ($\{\boldsymbol{s}_i^y\}$) and $N_D(z_1, z_2) = 100$ ($\{\boldsymbol{s}_j^{z_1}, \boldsymbol{s}_j^{z_2}\}$) available data in the reduced spaces of dimension $N_s(y) = 10$ and $N_s(z_1) + N_s(z_2) = 17$, respectively. Then, since $\boldsymbol{D}(y)$ and $\boldsymbol{D}(z_1, z_2)$ are generated independently, the overall PDF $f_{\boldsymbol{s}^{yz}}(\cdot)$ based on the entire set of $N_D(y) + N_D(z_1, z_2) = 200$ realizations is computed as $f_{\boldsymbol{s}^{yz}}(\cdot) = f_{\boldsymbol{s}^y}(\cdot) \cdot f_{\boldsymbol{s}^z}(\cdot)$ and employed in the characterization of epistemic uncertainty (Step 3). In particular, $N_e = 10000$ epistemic vectors $\boldsymbol{e}_l$, $l = 1, 2, \ldots, N_e$, are generated to uniformly probe E_0; for each $\boldsymbol{e}_l$, $N_a = 50000$ aleatory configurations are sampled in A to evaluate the average plausibilities $\mathcal{L}(\boldsymbol{e}_l)$. The final epistemic UM E is then defined as the smallest hyper-rectangle enveloping the joint 4-dimensional $\alpha \cdot 100\% = 95\%$ Confidence Interval (CI) of $\boldsymbol{e}$. The effectiveness of the proposed uncertainty quantification is testified by the considerable reduction (about 47%) in the volume of the epistemic space, i.e., from $Vol(E_0) = 2^4 = 16$ to $Vol(E) = 8.568$. Obviously, a more "aggressive" attitude (i.e., a smaller conservatism and a smaller value of α) would have led to a more relevant refinement. Notice that the explicit bounds of E are not reported here, as imposed by the Hosts of the NASA Challenge.

To build robustness in the quantification of aleatory uncertainty (Step 4), $N_e = 50$ epistemic points $\boldsymbol{e}_k$ (evenly distributed in E) are chosen, including $\boldsymbol{e}_{opt}$. For each $\boldsymbol{e}_k$, $k = 1, 2, \ldots, N_e$, $N_D(y) + N_D(z_1, z_2) = 200$ optimization problems (1) are

solved by GAs evolving a population of 50 possible solutions. The resulting aleatory vectors $\{\boldsymbol{a}_i: i = 1, 2, \ldots, N_D(y) + N_D(z_1, z_2)\}^k$ are fitted by SN distributions with polynomials of degree $d = 2$, to get N_e aleatory models $f_{\boldsymbol{A}}(\boldsymbol{a}|\boldsymbol{e}_k)$ of different plausibilities $\mathcal{L}(\boldsymbol{e}_k)$. For illustration purposes, Figure 1 shows the points $\{\boldsymbol{a}_i: i = 1, 2, \ldots, N_D(y) + N_D(z_1, z_2)\}^{opt}$ retrieved in correspondence of $\boldsymbol{e}_{opt}$ (red stars) and the SN-based level sets (CIs) 1-quantile apart (dashed lines) in the plane a_4-a_5. It is evident that SNs are very effective in characterizing the aleatory uncertainty by tightly enclosing the data and capturing complex dependencies and multi-modalities.

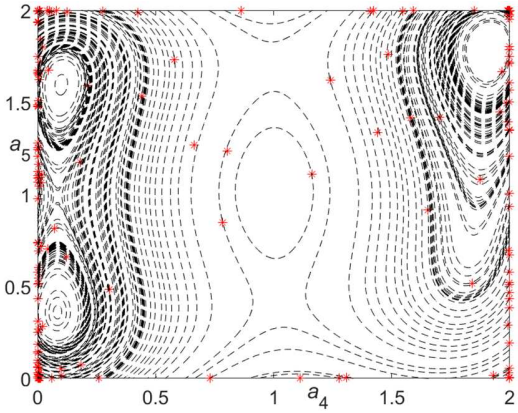

Figure 1. SN-based level sets of $f_{\boldsymbol{A}}(\boldsymbol{a}|\boldsymbol{e}_{opt})$ (lines) and retrieved input data (stars)

The results obtained at Step 4 provide evidence of the presence of non-negligible model discrepancies. In this view, Figure 2 shows the available reduced data set $\{\boldsymbol{s}_j^{z_2}: j = 1, 2, \ldots, N_D(z_1, z_2)\}$ (red stars) and the transformed output features $\{\boldsymbol{S}^{z_2}(z_2(\boldsymbol{a}_i, \boldsymbol{e}_{opt}, t)): i = 1, 2, \ldots, N_D(z_1, z_2)\}^{opt}$ (blue dots) resulting from the propagation of $\{\boldsymbol{a}_i: i = 1, 2, \ldots, N_D(y) + N_D(z_1, z_2)\}^{opt}$ and $\boldsymbol{e}_{opt}$ through the model $z_2(\cdot)$ and the successive dimensionality reduction by the corresponding SSAE (only the bi-dimensional projection $s_1^{z_2}$-$s_2^{z_2}$ of the feature space is shown for the sake of brevity). It is evident that, while the retrieved points (blue dots) capture well the aleatory distribution of the data (red stars), a systematic model error prevents an exact overlapping (similar behaviors are observable for models $y(\cdot)$ and $z_1(\cdot)$, not shown here for space limitations). Thus, the residuals $\boldsymbol{\delta}_{\boldsymbol{S}^y}(\boldsymbol{a}_i|\boldsymbol{e}_k)$, $i = 1, 2, \ldots, N_D(y) = 100$, $\boldsymbol{\delta}_{\boldsymbol{S}^{z_1}}(\boldsymbol{a}_j|\boldsymbol{e}_k)$ and $\boldsymbol{\delta}_{\boldsymbol{S}^{z_2}}(\boldsymbol{a}_j|\boldsymbol{e}_k)$, $j = 1, 2, \ldots, N_D(z_1, z_2) = 100$, are explicitly evaluated in the reduced space for all the aleatory models k constructed in correspondence of the different epistemic vectors $\boldsymbol{e}_k$, $k = 1, 2, \ldots, N_e = 50$. The discrepancy datasets thereby obtained are employed to train $N_s(y) + N_s(z_1) + N_s(z_2) = 10 + 13 + 4 = 27$ three-layered feed-forward ANNs with $n_a = 5$ inputs, one output and a number of hidden neurons optimized by Leave-One-Out (LOO) Cross-Validation (CV) due to the small amount of training data: values of the NRMSE smaller than 5.82% are obtained for all the features.

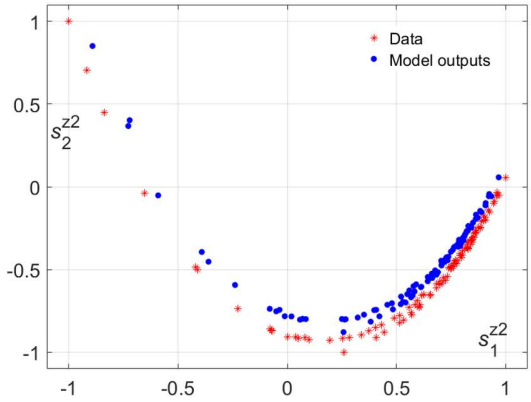

Figure 2. Model discrepancy for function z_2

The final UM $\langle f_{\boldsymbol{A}}(\boldsymbol{a}|\boldsymbol{e}_k), E\rangle$, with the corresponding discrepancy functions $\boldsymbol{\delta}_{\boldsymbol{S}^y}(\boldsymbol{a}_i|\boldsymbol{e}_k)$, $\boldsymbol{\delta}_{\boldsymbol{S}^{z_1}}(\boldsymbol{a}_j|\boldsymbol{e}_k)$ and $\boldsymbol{\delta}_{\boldsymbol{S}^{z_2}}(\boldsymbol{a}_j|\boldsymbol{e}_k)$ estimated in the reduced space for $k = 1, 2, \ldots, N_e = 50$, is propagated through the models $y(\cdot)$, $z_1(\cdot)$ and $z_2(\cdot)$ to verify the consistency of the results ($N_a = 10000$ aleatory samples are generated for each model k). For brevity and only for illustration purposes, Figure 3 (top) reports the time series observations $\{z_{2,i}(t)\}$ (red solid lines) along with the extreme upper and lower bounds (blue dashed lines) resulting from the UM propagation through function $z_2(\boldsymbol{a}, \boldsymbol{e}, t)$. The data is *captured* and *enveloped* quite tightly, at most of the time steps. Finally, in Figure 3 (bottom) the same calibration results are represented in the reduced space projected on the pair $s_1^{z_2}$-$s_2^{z_2}$: again, the calibrated model (i.e., the data enclosing set, shown as a blue solid line, and the level sets 1-quantile apart,

represented as black dashed lines) envelops the output data provided (red crosses). Based on these results, we can argue that the UM structure is likely to represent a "probability box" *possibly containing* (with the prescribed confidence α) the *true distribution* in the aleatory space. Also, the level of robustness and conservatism injected in the calibration process (see above) presumably allows the UM to properly withstand (i.e., "envelop") *most* aleatory and epistemic uncertainties, including *unexpected* and *extreme* events possibly occurring in the life of the system.

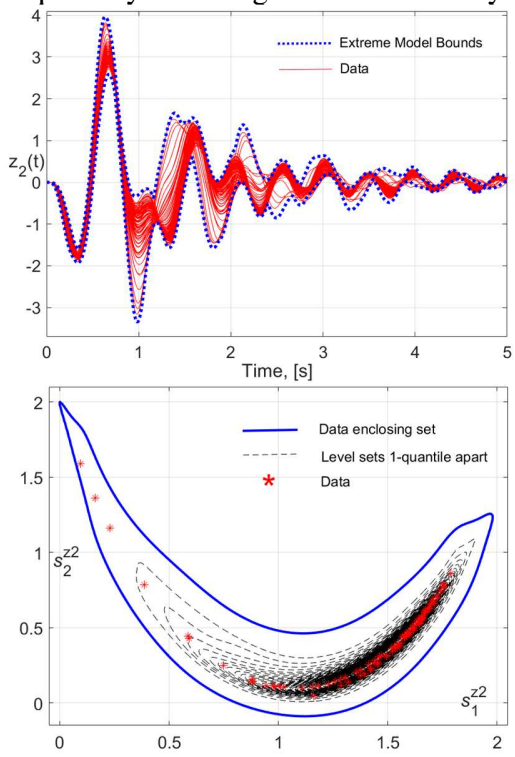

Figure 3. Calibrated model output z_2 against data

6. CONCLUSIONS

The paper has proposed an integrated approach for the inverse quantification of mixed aleatory (probabilistic) and epistemic (set-based) uncertainties in the high-dimensional models of dynamic systems, with sparse functional data. The effective combination of SSAEs, ANNs, GAs and SN distributions has been shown to perform satisfactorily on the 2020 NASA UQ Challenge: accurate and robust UMs (able to characterize complex, asymmetric and multi-modal dependencies together with model discrepancies) have been produced, while reducing the overall computational cost by two orders of magnitude.

The following issues and approaches are worth investigation in the future: (i) other methods to model complex dependences: e.g., copulas or non-parametric techniques based on Markov Chain Monte Carlo within a Bayesian framework; (ii) rigorous bounding methods (e.g., Interval Predictors and Scenario Theory) for model calibration with optimal control and elimination of outliers; (iii) adaptive metamodeling strategies in comparison to ANNs.

7. REFERENCES

Bi, S., Beer, M., Mottershead, J. (2022). Editorial: Recent advances in stochastic model updating, *Mechanical Systems and Signal Processing*, 172, 108971.

Crespo, L.G. , Colbert, B.K., Kenny, S.P., Giesy, D.P. (2019). On the quantification of aleatory and epistemic uncertainty using Sliced-Normal distributions, *Systems & Control Letters*, 134, 104560.

Crespo, L.G., and Kenny, S.P. (2021). The NASA Langley challenge on optimization under uncertainty. *Mechanical Systems and Signal Processing*, 152, 107405.

Dong, Z., Sheng, Z., Zhao, Y., Zhi, P. (2023). Robust optimization design method for structural reliability based on active-learning MPA-BP neural network, *Int. Journal of Structural Integrity*, doi: 10.1108/IJSI-10-2022-0129.

Maupin, K.A., and Swiler, L.P. (2020). Model discrepancy calibration across experimental settings, *Reliability Engineering & System Safety*, 200, 106818.

Nanty, S., Helbert, C., Marrel, A., Pérot, N., Prieur, C. (2017). Uncertainty quantification for functional dependent random variables, *Computational Statistics*, 32(2), 559-583.

Wu, X., Xie, Z., Alsafadi, F., Kozlowski, T. (2021). A comprehensive survey of inverse uncertainty quantification of physical model parameters in nuclear system thermal–hydraulics codes, *Nuclear Engineering and Design*, 384, 111460.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., Gao RX. (2019). Deep learning and its applications to machine health monitoring. *Mech Syst Signal Process*, 115, 213–237.