Neural network hyperreduction for parameterized computational models in nonlinear stochastic dynamics

Olivier Ezvan, C. Desceliers, Christian Soize, E. Capiez-Lernout

Neural network hyperreduction for parameterized computational models in nonlinear stochastic dynamics

O. Ezvan[a,*], C. Desceliers[a], C. Soize[a], E. Capiez-Lernout[a]

[a] *Univ Gustave Eiffel, MSME UMR 8208 CNRS, 5 Bd Descartes, 77454 Marne-La-Vallée Cedex 02, France*

Abstract

Highly efficient hyperreduction methods have been proposed to reduce the computational cost of parameterized high-dimensional models in nonlinear dynamics. Unlike hyperreduced models, the parameterized reduced-order models still scale with the dimension of the high-dimensional model for the computation of the projected nonlinear forces. As an alternative to traditional hyperreduction methods, this work presents a neural-network-based hyperreduction strategy. It is embedded within the framework of parameterized reduced-order models in nonlinear stochastic dynamics. The proposed approach achieves hyperreduction by entirely removing such a dependence on the high-dimensional model dimension. In particular, the costly computation and projection of nonlinear forces in the high-dimensional physical space is bypassed: the neural network directly outputs the reduced forces from the reduced displacements and system parameters. It is trained on data collected during the offline stage required for constructing the parameterized reduced-order model. The resulting hyperreduced model enables substantial computational savings, particularly suited for large-scale Monte Carlo simulations in the stochastic dynamics framework. Numerical results show that the method provides highly accurate approximations of the nonlinear response and its statistics, with a performance gain of two orders of magnitude with respect to the parameterized reduced-order model.

Keywords: hyperreduction, nonlinear stochastic dynamics, reduced-order models, uncertainty quantification, neural networks

1. List of symbols

Convention for the variables, vectors, and matrices.
x, η: lower-case Latin or Greek letters are deterministic real variables.
$\mathbf{x}, \boldsymbol{\eta}$: boldface lower-case Latin or Greek letters are deterministic vectors.
X: upper-case Latin letters are real-valued random variables.
$\mathbf{X}$: boldface upper-case Latin letters are vector-valued random variables.
$[x]$: lower-case Latin letters between brackets are deterministic matrices.

Published online in *Computational Mechanics* on February 14, 2026. DOI: 10.1007/s00466-025-02739-6.
*Corresponding author.
Email addresses: olivier.ezvan@univ-eiffel.fr (O. Ezvan), christophe.desceliers@univ-eiffel.fr (C. Desceliers), christian.soize@univ-eiffel.fr (C. Soize), evangeline.capiez-lernout@univ-eiffel.fr (E. Capiez-Lernout)

[**X**]: boldface upper-case letters between brackets are matrix-valued random variables.

Algebraic notations and acronyms.
$\mathbb{R}$: set of real numbers.
$\mathbb{R}^\nu$: Euclidean vector space of dimension ν.
$\mathbb{M}_{n,m}$, $\mathbb{M}_n$: set of the $(n \times m)$, $(n \times n)$, real matrices.
$\mathbb{M}_n^+$: set of positive-definite $(n \times n)$ real matrices.

ANN: artificial neural network.
DOF: degree of freedom.
FE: finite element.
FRF: frequency response function.
GKDE: Gaussian kernel density estimation.
HDM: high-dimensional model.
HROM: hyperreduced model.
MC: Monte-Carlo.
PCA: principal component analysis.
PDF: probability density function.
POD: proper orthogonal decomposition.
RMSE: root mean-square error.
ROM: reduced-order model.
SVD: singular value decomposition.

Convention used for random variables.
In this paper, for any finite integer $m \geq 1$, the Euclidean space $\mathbb{R}^m$ is equipped with the σ-algebra $\mathcal{B}_{\mathbb{R}^m}$. If $\mathbf{Y}$ is a $\mathbb{R}^m$-valued random variable defined on the probability space $(\Theta, \mathcal{T}, \mathcal{P})$, $\mathbf{Y}$ is a mapping $\theta \mapsto \mathbf{Y}(\theta)$ from Θ into $\mathbb{R}^m$, measurable from $(\Theta, \mathcal{T})$ into $(\mathbb{R}^m, \mathcal{B}_{\mathbb{R}^m})$, and $\mathbf{Y}(\theta)$ is a realization of $\mathbf{Y}$ for $\theta \in \Theta$. The probability distribution of $\mathbf{Y}$ is the probability measure $P_{\mathbf{Y}}(d\mathbf{y})$ on the measurable set $(\mathbb{R}^m, \mathcal{B}_{\mathbb{R}^m})$ (we will simply say on $\mathbb{R}^m$). The Lebesgue measure on $\mathbb{R}^m$ is noted $d\mathbf{y}$ and when $P_{\mathbf{Y}}(d\mathbf{y})$ is written as $p_{\mathbf{Y}}(\mathbf{y})\, d\mathbf{y}$, $p_{\mathbf{Y}}$ is the probability density function (PDF) on $\mathbb{R}^m$ of $P_{\mathbf{Y}}(d\mathbf{y})$ with respect to $d\mathbf{y}$. Finally, E denotes the mathematical expectation operator that is such that $E\{\mathbf{Y}\} = \int_{\mathbb{R}^m} \mathbf{y}\, P_{\mathbf{Y}}(d\mathbf{y})$.

2. Introduction

Context. Recent advances in reduced-order models and surrogate-based hyperreduction techniques have provided robust foundations for reducing the high numerical cost of complex computational models. For instance, machine learning-based approaches such as kernel and operator learning [1, 2, 3] and data-driven modeling via neural networks [4, 5] have allowed for capturing nonlinear behavior in complex systems. Works that explore graph-based surrogates [6, 7], convolutional neural networks-driven multiscale frameworks [8, 9], and adaptive stochastic reduced-order models (ROM) [10, 11] integrate physics-informed neural architectures. The development of recurrent neural networks for dynamics [12, 13] and probabilistic learning for uncertainty-aware design [14, 15, 16] achieves computational efficiency under uncertainty.

Projection-based model order reduction techniques are widely used to reduce the computational cost of parameterized large-scale simulations involving nonlinear time-dependent complex mod-

els [17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. The parameterized ROM is obtained by projecting the high-dimensional model (HDM) onto a low-dimensional subspace identified via Proper Orthogonal Decomposition (POD) using Singular Value Decomposition (SVD), and based on snapshots of the nonlinear responses computed by the HDM for all the sampled values of the parameters. Such a parameterized ROM retains the key dynamic features of the original system while enabling significant computational acceleration [27, 28], as the vector-valued parameter varies over its admissible set. However, when applied to nonlinear problems, the ROM still requires the evaluation and projection of nonlinear terms at each time step and iteration, operations whose cost scales with the size of the HDM. To further reduce this cost, hyperreduction techniques [29, 17, 19, 30, 31, 32] have been developed. In the context of stochastic dynamics, the Monte Carlo (MC) method remains a standard for uncertainty propagation. Yet, its reliance on thousands of time-domain simulations makes the cost prohibitive when the HDM or even the ROM is too expensive to evaluate. This challenge motivates the need for hyperreduced models (HROM) that eliminate any dependence on the HDM size during online evaluation.

Novelty. This work presents a hyperreduction strategy based on Artificial Neural Networks (ANN). This hyperreduction is not the construction of a statistical surrogate model whose inputs are the system parameters and the outputs are the observed responses of the nonlinear stochastic dynamical model. Instead, the proposed hyperreduction consists in constructing a deterministic surrogate model, whose inputs are defined on the product space of both the reduced (generalized) displacements of the ROM and of the system parameters, and whose outputs are defined on the space of the reduced (projected) nonlinear internal forces. Regarding ANN-based hyperreduction, the concept was first introduced in [33] for a one-dimensional nonlinear thermal problem. Independently, related work addressed first-order partial differential equations in computational fluid dynamics [34], where a single network learns the reduced velocity function in a parameterized ROM as a function of time, jointly capturing internal nonlinear effects and external loading. A related approach has been applied to three-dimensional elasticity [35], learning the time-dependent reduced residual and Jacobian, with time as an input and without using the reduced displacements as inputs. Finally, ANN-based hyperreduction has been explored for parameterized static nonlinear elasticity using physics-augmented networks enforcing material consistency, stability, and hyperelasticity [36]. In contrast to the cited papers, the present work directly learns the reduced nonlinear forces as a function of the reduced displacements (as opposed to the solution as a function of time) and tackles large-scale structural dynamics in a stochastic setting. The methodology proposed in this paper can be summarized as follows. The ANN is trained on data collected during the offline construction of the parameterized ROM, a phase that is anyway required. Once trained, the ANN surrogate replaces the nonlinear force evaluation and projection, yielding an HROM that is fully independent of the HDM dimension during the online phase. The proposed ANN-based HROM is particularly well suited for large-scale MC simulations, where a massive number of realizations must be generated at low cost. Numerical results demonstrate that the method provides highly accurate approximations of the nonlinear response and its statistics, with performance gains exceeding two orders of magnitude compared to the reference ROM.

Organization of the paper. The paper is organized as follows. Section 3 introduces the high-dimensional model and its parameterized reduced-order counterpart. Section 4 presents the ANN-based hyperreduction strategy. Section 5 details the ANN training procedure. Section 6 provides numerical results and performance evaluations. Concluding remarks are drawn in Sec-

tion 7. Three appendices are provided. Appendix A describes the preprocessing of the training data. Appendix B compares the linear and nonlinear responses to highlight the importance of nonlinear effects. Using the HROM to generate a large number of MC realizations, Appendix C compares nonparametric conditional estimation with direct estimation at a fixed control-parameter value, showing that the proposed approach achieves statistically converged stochastic responses at a small fraction of the computational cost that an equivalent ROM computation would require.

3. Parameterized computational model for nonlinear stochastic dynamics

First, the deterministic high-dimensional nonlinear dynamical model is presented in Section 3.1. This model involves two types of parameters: the control parameter $\mathbf{w}$ and the uncontrolled parameter $\mathbf{u}$. By introducing randomness into $\mathbf{u}$, modeled as a random vector $\mathbf{U}$, the corresponding nonlinear stochastic dynamical model is obtained, as described in Section 3.2. Then, we present the construction of the training dataset that allows the stochastic ROM to be obtained.

3.1. Deterministic high-dimensional nonlinear dynamical model

The equation of motion is discretized using the finite element (FE) method, and the FE solution vector of displacements $\mathbf{y}$ is the solution of the following nonlinear differential equation in matrix form for the HDM in the time domain,

$$[\mathbb{M}(\mathbf{w},\mathbf{u})]\,\ddot{\mathbf{y}}(t;\mathbf{w},\mathbf{u}) + [\mathbb{D}(\mathbf{w},\mathbf{u})]\,\dot{\mathbf{y}}(t;\mathbf{w},\mathbf{u}) + [\mathbb{K}(\mathbf{w},\mathbf{u})]\,\mathbf{y}(t;\mathbf{w},\mathbf{u}) + \mathbf{f}^{\text{NL}}\left(\mathbf{y}(t;\mathbf{w},\mathbf{u});\mathbf{w},\mathbf{u}\right) = \mathbf{f}^{\text{exc}}(t;\mathbf{w},\mathbf{u})\,, \tag{1}$$

in which $t \in T = [t_i, t_f]$ is the time, and where $\mathbf{w} \in C_w \subset \mathbb{R}^{n_w}$ is the control parameter (design parameter) and $\mathbf{u} \in C_u \subset \mathbb{R}^{n_u}$ is the uncontrolled parameter (latent uncertainty parameter). In this equation, $[\mathbb{M}(\mathbf{w},\mathbf{u})]$, $[\mathbb{D}(\mathbf{w},\mathbf{u})]$, and $[\mathbb{K}(\mathbf{w},\mathbf{u})]$ represent the FE mass, damping, and stiffness matrices, which are $(n_{\text{FE}} \times n_{\text{FE}})$ positive-definite matrices, with n_{FE} the number of degrees of freedom (DOF) of the FE model. The vector $\mathbf{f}^{\text{NL}} \in \mathbb{R}^{n_{\text{FE}}}$ represents the nonlinear forces with respect to displacements $\mathbf{y} \in \mathbb{R}^{n_{\text{FE}}}$ (geometric nonlinearities). The vector $\mathbf{f}^{\text{exc}} \in \mathbb{R}^{n_{\text{FE}}}$ represents the excitation forces applied to the structure. In general, all these quantities depend on both control and uncontrolled parameters $\mathbf{w}$ and $\mathbf{u}$.

3.2. Stochastic high-dimensional nonlinear dynamical model

The uncontrolled parameter $\mathbf{u}$ is modeled as a random variable $\mathbf{U} \in C_u$ with given PDF on $\mathbb{R}^{n_u}$ whose support is C_u. The random matrices $[\mathbb{M}(\mathbf{w},\mathbf{U})]$, $[\mathbb{D}(\mathbf{w},\mathbf{U})]$, and $[\mathbb{K}(\mathbf{w},\mathbf{U})]$ are rewritten as $[\mathbf{M}(\mathbf{w})]$, $[\mathbf{D}(\mathbf{w})]$, and $[\mathbf{K}(\mathbf{w})]$. Similarly, the vector-valued random functions $\mathbf{f}^{\text{NL}}(\cdot\,,\mathbf{w},\mathbf{U})$ and $\mathbf{f}^{\text{exc}}(\cdot\,;\mathbf{w},\mathbf{U})$ are rewritten as $\mathbf{F}^{\text{NL}}(\cdot\,;\mathbf{w})$ and $\mathbf{F}^{\text{exc}}(\cdot\,;\mathbf{w})$. With this notation, the stochastic HDM associated with Eq. (1) is written as

$$[\mathbf{M}(\mathbf{w})]\,\ddot{\mathbf{Y}}(t;\mathbf{w}) + [\mathbf{D}(\mathbf{w})]\,\dot{\mathbf{Y}}(t;\mathbf{w}) + [\mathbf{K}(\mathbf{w})]\,\mathbf{Y}(t;\mathbf{w}) + \mathbf{F}^{\text{NL}}\left(\mathbf{Y}(t;\mathbf{w});\mathbf{w}\right) = \mathbf{F}^{\text{exc}}(t;\mathbf{w})\,, \tag{2}$$

in which $[\mathbf{M}(\mathbf{w})]$, $[\mathbf{D}(\mathbf{w})]$, and $[\mathbf{K}(\mathbf{w})]$ are random matrices with values in $\mathbb{M}^+_{n_{\text{FE}}}$. The vector-valued random variable $\mathbf{Y}(\cdot\,;\mathbf{w})$ represents the solution displacements. This stochastic HDM depends on control parameter $\mathbf{w}$. In the context of robust optimization, parameter $\mathbf{w}$ is identified to fit the HDM random response to predefined targets under uncertainty.

3.3. Training stage using the parameterized stochastic high-dimensional model

The control parameter $\mathbf{w}$ is also modeled by a random variable $\mathbf{W}$, with given PDF on $\mathbb{R}^{n_w}$ with support C_w. The independent realizations of the random variables $\mathbf{W}$ and $\mathbf{U}$ are generated using their PDF. We denote by $\mathbf{w}^j$ and $\mathbf{u}^j$ the j-th realization of $\mathbf{W}$ and $\mathbf{U}$. Solving Eq. (2) by the MC simulation method with these realizations of $\mathbf{W}$ and $\mathbf{U}$, yields the training dataset $\mathcal{D}_{\text{train}}$ such that

$$\mathcal{D}_{\text{train}} = \{\mathbf{w}^j, \mathbf{u}^j, [\,y^j\,]\}_{j=1,\ldots,n_{\text{train}}} . \tag{3}$$

In Eq. (3), the matrix $[\,y^j\,]$ denotes the j-th realization of the time response matrix, such that

$$[\,y^j\,] = \left[\mathbf{y}(t_1; \mathbf{w}^j, \mathbf{u}^j) \ldots \mathbf{y}(t_{N_t}; \mathbf{w}^j, \mathbf{u}^j)\right] \in \mathbb{M}_{n_{\text{FE}},N_t} , \tag{4}$$

in which $t_1, \ldots, t_{N_t}$ result from the discretization of T with N_t time steps. This training dataset $\mathcal{D}_{\text{train}}$ is used to construct the stochastic ROM presented in Section 3.4. The collection of all temporal responses $[\,y^j\,]$ for the n_{train} realizations is used to form the snapshot matrix $[\,Y\,]$, where each column corresponds to a system response at a specific time step and a specific realization of the random parameters. This snapshot matrix is written as

$$[\,Y\,] = \left[[\,y^1\,] \ldots [\,y^{n_{\text{train}}}\,]\right] \in \mathbb{M}_{n_{\text{FE}},n_{\text{snap}}} , \tag{5}$$

where n_{snap} denotes the number of response snapshots, given in the present case by $n_{\text{snap}} = N_t \times n_{\text{train}}$ (more generally, not all time steps are necessarily retained). The columns of $[\,Y\,]$ are used to span a vector subspace of reduced dimension, in which the dynamical response $t \mapsto \mathbf{y}(t; \mathbf{w}, \mathbf{u})$, solution of Eq. (1), can be approximated for any $\mathbf{w} \in C_w$ and $\mathbf{u} \in C_u$.

3.4. Parameterized stochastic reduced-order model

The SVD of the centered response is considered,

$$[\Phi][\Sigma][V]^T = \frac{1}{\sqrt{n_{\text{snap}} - 1}} \left([\,Y\,] - [\,\overline{Y}\,]\right) , \tag{6}$$

in which each column of matrix $[\,\overline{Y}\,]$ is the same vector, which is defined as the average of all the columns of $[\,Y\,]$. In Eq. (6), $[\Phi]$ is the matrix of the left-singular vectors, $[V]$ is the matrix of the right-singular vectors, and $[\Sigma]$ is the diagonal matrix of the singular values. Upon truncation, matrix $[\Phi]$ is with size $(n_{\text{FE}} \times n_{\text{POD}})$, in which n_{POD} is the number of kept POD modes. Matrix $[\Phi]$ satisfies $[\Phi]^T[\Phi] = [I_{n_{\text{POD}}}]$, where $[I_{n_{\text{POD}}}]$ denotes the identity matrix of dimension n_{POD}. Let ε_{POD} be the POD error due to the truncation such that

$$\varepsilon_{\text{POD}}(n_{\text{POD}}) = 1 - \frac{\sum_{k=1}^{n_{\text{POD}}} \sigma_k^2}{\sum_{k=1}^{n_{\text{snap}}} \sigma_k^2} , \tag{7}$$

in which σ_k is the k-th singular value, with the singular values sorted in descending order. In practice, only a small number of singular values is computed, and the term $\sum_{k=1}^{n_{\text{snap}}} \sigma_k^2$ is evaluated by $\frac{1}{n_{\text{snap}}} \sum_i \sum_j ([\,Y\,] - [\,\overline{Y}\,])_{ij}^2$. Reduced dimension $n_{\text{POD}} \leq \min(n_{\text{snap}}, n_{\text{FE}})$ is the minimum number n_{POD} such that $\varepsilon_{\text{POD}}(n_{\text{POD}}) \leq \varepsilon_{\text{tol}}$, with ε_{tol} a given error tolerance. The POD basis is given by the columns of matrix $[\Phi]$ and is used to project the dynamical equation. The parameterized

stochastic ROM is thus obtained by Galerkin projection, such that

$$([\Phi]^T[\mathbf{M}(\mathbf{w})][\Phi])\ddot{\mathbf{Q}}(t;\mathbf{w}) + ([\Phi]^T[\mathbf{D}(\mathbf{w})][\Phi])\dot{\mathbf{Q}}(t;\mathbf{w}) + ([\Phi]^T[\mathbf{K}(\mathbf{w})][\Phi])\mathbf{Q}(t;\mathbf{w}) + [\Phi]^T\mathbf{F}^{\text{NL}}([\Phi]\mathbf{Q}(t;\mathbf{w}),\mathbf{w}) = [\Phi]^T\mathbf{F}^{\text{exc}}(t;\mathbf{w}), \tag{8}$$

in which $\mathbf{Q}$ is an $\mathbb{R}^{n_{\text{POD}}}$-valued random variable of reduced (generalized) displacements, such that the $\mathbb{R}^{n_{\text{FE}}}$-valued random variable $\mathbf{Y}$ of physical displacements can be approximated (keeping the same notation $\mathbf{Y}$) as

$$\mathbf{Y} = [\Phi]\mathbf{Q}. \tag{9}$$

The dimension n_{POD} of the parameterized random matrix equation Eq. (8) satisfies $n_{\text{POD}} \ll n_{\text{FE}}$ and can therefore be solved, *a priori*, at a reduced computational cost. For a sufficiently large number n_{train} of realizations of the HDM and provided adequate sampling, the stochastic ROM can be considered valid for all $\mathbf{w} \in C_w$ and for all $\mathbf{u} \in C_u$.

4. Neural network hyperreduction

In this Section, we begin with presenting the hyperreduction framework, followed by the proposed neural network hyperreduction strategy.

4.1. Hyperreduction framework

We now consider that $\mathbf{y}$ and $\mathbf{q}$ are functions of t, $\mathbf{w}$, and $\mathbf{u}$ and that $\mathbf{f}^{\text{NL}}$ is a function of $\mathbf{y}$, $\mathbf{w}$, and $\mathbf{u}$. Let $\boldsymbol{\mathcal{F}}^{\text{NL}} = [\Phi]^T\mathbf{f}^{\text{NL}}$ with values in $\mathbb{R}^{n_{\text{POD}}}$ denote the deterministic function of reduced nonlinear forces, associated with deterministic $\mathbb{R}^{n_{\text{FE}}}$-valued function $\mathbf{f}^{\text{NL}}$ of physical nonlinear forces in Eq. (1). Function $\mathbf{f}^{\text{NL}}$ depends on the function $\mathbf{y}$ of displacements as well as on parameters $\mathbf{w}$ and $\mathbf{u}$. Therefore, function $\boldsymbol{\mathcal{F}}^{\text{NL}}$ depends on the reduced displacements $\mathbf{q}$ such that $\mathbf{y} = [\Phi]\mathbf{q}$ as well as on the parameters $\mathbf{w}$ and $\mathbf{u}$. This writes as follows,

$$\boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q},\mathbf{w},\mathbf{u}) = [\Phi]^T\mathbf{f}^{\text{NL}}([\Phi]\mathbf{q},\mathbf{w},\mathbf{u}). \tag{10}$$

The use of the stochastic ROM requires evaluations of vector $\boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})$ for given values of $\mathbf{q}$, $\mathbf{w}$, and $\mathbf{u}$, which is computationally expensive. Indeed, it requires computing the high-dimensional nonlinear force vector $\mathbf{f}^{\text{NL}}([\Phi]\mathbf{q},\mathbf{w},\mathbf{u}) \in \mathbb{R}^{n_{\text{FE}}}$, followed by the projection step $[\Phi]^T\mathbf{f}^{\text{NL}}([\Phi]\mathbf{q},\mathbf{w},\mathbf{u}) \in \mathbb{R}^{n_{\text{POD}}}$. This evaluation must be performed:

- for each MC realization,
- at each time step of the realization (N_t time steps),
- at each iteration of the nonlinear solution algorithm, such as Newton-Raphson algorithm and adapted [37] arc-length algorithm [38, 39].

The cost of an evaluation is dependent on the dimension n_{FE} of the HDM. The goal of hyperreduction is to remove this dependency with respect to n_{FE}, so that the computational cost of the stochastic ROM scales only with its reduced dimension n_{POD}.

It should be noted that, for the Newton–Raphson algorithm devoted to the iterative solution of the nonlinear equations, the construction of the tangent stiffness matrix is required. Within the use of the stochastic ROM, it is the reduced tangent stiffness matrix that is required, which can

be obtained with the Jacobian matrix $[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})] \in \mathbb{M}_{n_{\mathrm{POD}}}$, where $\mathrm{D}_{\mathbf{q}}$ is the differential operator such that

$$[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})]_{ij} = \frac{\partial}{\partial q_j}\mathcal{F}_i^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})\,, \tag{11}$$

where $\mathcal{F}_i^{\mathrm{NL}}$ is the i-th component of $\boldsymbol{\mathcal{F}}^{\mathrm{NL}}$ and q_j is the j-th component of $\mathbf{q}$. The computational cost involved in computing such Jacobian matrix is also (indirectly) dependent on the dimension n_{FE} of the HDM.

4.2. Neural network hyperreduction

The strategy proposed is to replace the nonlinear application $(\mathbf{q},\mathbf{w},\mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})$ by an ANN surrogate model $(\mathbf{q},\mathbf{w},\mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u})$ such that

$$\boldsymbol{\mathcal{F}}^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u}) \simeq \boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})\,, \tag{12}$$

for which the accuracy has to be controlled (see Section 5.3). As a requirement, the surrogate must remain sufficiently accurate for all $(\mathbf{w},\mathbf{u}) \in C_w \times C_u$ and for any reduced displacement state $\mathbf{q} \in \mathbb{R}^{n_{\mathrm{POD}}}$ that may be encountered in subsequent dynamical simulations. Once the surrogate is constructed, the Jacobian matrix $[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})]$ can be computed efficiently either by finite differences or by reverse-mode automatic differentiation (neural-network backpropagation). To this end, the matrix $[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})]$ is replaced with its ANN-based approximation $[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u})]$ such that

$$[\mathrm{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u})]_{ij} = \frac{\partial}{\partial q_j}\mathcal{F}_i^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u})\,, \tag{13}$$

where $\mathcal{F}_i^{\mathrm{ANN}}$ is the i-th component of $\boldsymbol{\mathcal{F}}^{\mathrm{ANN}}$.

5. Neural network training

This section encompasses the data collection stage, the training data preprocessing, and the neural network training stage, outlining the end-to-end process of preparing and training the ANN.

5.1. Data collection stage

To train the ANN surrogate $(\mathbf{q},\mathbf{w},\mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\mathrm{ANN}}(\mathbf{q},\mathbf{w},\mathbf{u})$, it is necessary to collect a large number of samples of the input $(\mathbf{q},\mathbf{w},\mathbf{u})$ and associated output $\boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q},\mathbf{w},\mathbf{u})$. To do so, the training stage described in Section 3.3 is considered. The dataset $\mathcal{D}_{\mathrm{ANN}}$, referred to as the ANN full training dataset, is defined as

$$\mathcal{D}_{\mathrm{ANN}} = \{\boldsymbol{\mathcal{F}}^{\mathrm{NL},i}, \mathbf{q}^i, \mathbf{w}^i, \mathbf{u}^i\}_{i=1,\dots,N_{\mathrm{data}}}\,, \tag{14}$$

with $\boldsymbol{\mathcal{F}}^{\mathrm{NL},i} = \boldsymbol{\mathcal{F}}^{\mathrm{NL}}(\mathbf{q}^i,\mathbf{w}^i,\mathbf{u}^i)$ and where $N_{\mathrm{data}} = n_{\mathrm{train}} N_t \nu_{\mathrm{iter}}$ denotes the number of training data points for the ANN, with ν_{iter} being the average number of iterations of the nonlinear solution algorithm per time step.

However, the vectors $\{\boldsymbol{\mathcal{F}}^{\mathrm{NL},i}\}_i$ and $\{\mathbf{q}^i\}_i$ to be collected are reduced vectors of dimension n_{POD}, whereas the training stage described in Section 3.3 involves the HDM, from which their high-dimensional counterparts $\{\mathbf{f}^{\mathrm{NL},i}\}_i$ and $\{\mathbf{y}^i\}_i$ (dimension n_{FE}) are computed during the nonlinear

iterations. Two approaches can therefore be considered to construct the desired ANN full training dataset $\mathcal{D}_{\text{ANN}}$ from Eq. (14), as follows:

- *Option 1*: During the training stage described in Section 3.3, store the physical quantities $\{\mathbf{f}^{\text{NL},i}\}_i$ and $\{\mathbf{y}^i\}_i$ on disk. Then, once the POD basis $[\Phi]$ has been constructed, compute the reduced vectors via projection: $\boldsymbol{\mathcal{F}}^{\text{NL},i} = [\Phi]^T \mathbf{f}^{\text{NL},i}$ and $\mathbf{q}^i = [\Phi]^T \mathbf{y}^i$.
- *Option 2*: If storing the physical vectors is not feasible, repeat the entire training stage using the stochastic ROM instead of the stochastic HDM. This allows for the direct collection of the reduced quantities $\{\boldsymbol{\mathcal{F}}^{\text{NL},i}\}_i$ and $\{\mathbf{q}^i\}_i$, along with the system parameters $\{\mathbf{w}^i\}_i$ and $\{\mathbf{u}^i\}_i$.

Option 1 requires writing the high-dimensional displacement and nonlinear force vectors to hard disk at each Newton iteration of every time step and retaining them on persistent storage. While write operations add overhead to the offline phase, the storage requirement is the dominant bottleneck and quickly becomes prohibitive for large-scale HDM and many realizations. By contrast, *Option 2* avoids HDM-scale storage by generating the dataset with the ROM, at the cost of a moderate offline runtime overhead as in general, the ROM runs faster than the HDM.

5.2. Training data preprocessing

Prior to training the ANN, the dataset $\mathcal{D}_{\text{ANN}}$ is preprocessed in order to facilitate the training stage. The input parameters are projected onto their principal component analysis (PCA) basis so as to obtain fewer, uncorrelated, and normalized input parameters. The processing is detailed in Appendix A, in the framework of the numerical application presented in Section 6.

5.3. Neural network training stage

The ANN full training dataset $\mathcal{D}_{\text{ANN}}$ is partitioned into three disjoint subsets $\mathcal{D}_{\text{ANN},1}$, $\mathcal{D}_{\text{ANN},2}$, and $\mathcal{D}_{\text{ANN},3}$ such that

$$\mathcal{D}_{\text{ANN}} = \mathcal{D}_{\text{ANN},1} \cup \mathcal{D}_{\text{ANN},2} \cup \mathcal{D}_{\text{ANN},3} \,, \tag{15}$$

which correspond to a partitioning of the n_{train} realizations into three disjoint groups of $n_{\text{train},1}$, $n_{\text{train},2}$, and $n_{\text{train},3}$ realizations respectively, such that

$$n_{\text{train}} = n_{\text{train},1} + n_{\text{train},2} + n_{\text{train},3} \,. \tag{16}$$

The three subsets $\mathcal{D}_{\text{ANN},1}$, $\mathcal{D}_{\text{ANN},2}$, and $\mathcal{D}_{\text{ANN},3}$ include $N_{\text{data},1}$, $N_{\text{data},2}$, and $N_{\text{data},3}$ data points, respectively, such that

$$N_{\text{data}} = N_{\text{data},1} + N_{\text{data},2} + N_{\text{data},3} \,. \tag{17}$$

The ANN full dataset is thus divided into three subsets: the ANN training subset with $N_{\text{data},1}$ data points, the ANN validation subset with $N_{\text{data},2}$ data points, and the ANN test subset with $N_{\text{data},3}$ data points. It should be noted that for $j = 1, 2, 3$, the number $N_{\text{data},j}$ of data points of subset $\mathcal{D}_{\text{train},j}$ comes from $n_{\text{train},j}$ MC realizations, and is such that

$$N_{\text{data},j} = n_{\text{train},j}\, N_t\, \nu_{\text{iter},j} \,, \tag{18}$$

in which, similarly to ν_{iter} , the quantity $\nu_{\text{iter},j}$ denotes the average number of iterations of the nonlinear solution algorithm for each time step, restricted to the $n_{\text{train},j}$ realizations of subset $\mathcal{D}_{\text{train},j}$.

To train the ANN, its weights and biases are optimized using the training subset $\mathcal{D}_{\text{ANN},1}$ of $N_{\text{data},1}$ data points. The training subset $\mathcal{D}_{\text{ANN},1}$ is partitioned into n_{MB} mini-batches, and the optimization is performed mini-batch by mini-batch. For a given mini-batch, the following cost function $J(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the weights and biases, is minimized:

$$J(\boldsymbol{\theta}) = \sqrt{\frac{\frac{1}{n_{\text{POD}}}\sum_{k=1}^{n_{\text{POD}}}\frac{1}{N_{\text{mb}}}\sum_{\ell=1}^{N_{\text{mb}}}\left(\mathcal{F}_k^{\text{ANN},\ell}(\boldsymbol{\theta}) - \mathcal{F}_k^{\text{NL},\ell}\right)^2}{\frac{1}{n_{\text{POD}}}\sum_{k=1}^{n_{\text{POD}}}\frac{1}{N_{\text{mb}}}\sum_{\ell=1}^{N_{\text{mb}}}\left(\mathcal{F}_k^{\text{NL},\ell}\right)^2}}, \tag{19}$$

in which N_{mb} denotes the number of data points in the considered mini-batch, and where $\mathcal{F}_k^{\text{NL},\ell}$ (resp. $\mathcal{F}_k^{\text{ANN},\ell}(\boldsymbol{\theta})$) denotes the k-th component of the ℓ-th vector $\boldsymbol{\mathcal{F}}^{\text{NL}}$ (resp. $\boldsymbol{\mathcal{F}}^{\text{ANN}}(\boldsymbol{\theta})$) in the considered mini-batch. The cost function $J(\boldsymbol{\theta})$ is a normalized root mean square error (normalized RMSE), which corresponds to a normalized quadratic error calculated over the current mini-batch.

An epoch is defined as one complete pass through the entire training subset $\mathcal{D}_{\text{ANN},1}$, during which all mini-batches are processed once. That is, the first epoch is completed once all the mini-batches have been covered. Once an epoch is completed, the accuracy of the ANN is evaluated. The accuracy is denoted as $A_{\text{val}}(\boldsymbol{\theta})$ and is defined as

$$A_{\text{val}}(\boldsymbol{\theta}) = 1 - \sqrt{\frac{\frac{1}{n_{\text{POD}}}\sum_{k=1}^{n_{\text{POD}}}\frac{1}{N_{\text{data},2}}\sum_{\ell=1}^{N_{\text{data},2}}\left(\mathcal{F}_k^{\text{ANN},\ell}(\boldsymbol{\theta}) - \boldsymbol{\mathcal{F}}_k^{\text{NL},\ell}\right)^2}{\frac{1}{n_{\text{POD}}}\sum_{k=1}^{n_{\text{POD}}}\frac{1}{N_{\text{data},2}}\sum_{\ell=1}^{N_{\text{data},2}}\left(\boldsymbol{\mathcal{F}}_k^{\text{NL},\ell}\right)^2}}, \tag{20}$$

in which it is recalled that $N_{\text{data},2}$ is the number of data points in the validation subset $\mathcal{D}_{\text{train},2}$, and where $\mathcal{F}_k^{\text{NL},\ell}$ (resp. $\mathcal{F}_k^{\text{ANN},\ell}(\boldsymbol{\theta})$) denotes the k-th component of the ℓ-th vector $\boldsymbol{\mathcal{F}}^{\text{NL}}$ (resp. $\boldsymbol{\mathcal{F}}^{\text{ANN}}(\boldsymbol{\theta})$) in the validation subset. The notation $A_{\text{val}}(\boldsymbol{\theta})$ stands for validation, as it is evaluated over the entire validation subset $\mathcal{D}_{\text{ANN},2}$ of $N_{\text{data},2}$ data points. That is, the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ is relative to data unseen by the optimization process. It should be noted that similarly to cost function $J(\boldsymbol{\theta})$, the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ is obtained from the normalized quadratic error. Once an epoch is completed and its accuracy $A_{\text{val}}(\boldsymbol{\theta})$ is evaluated, the entire process is repeated, which constitutes the second epoch. As many epochs as necessary are performed, until the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ reaches a desired level.

6. Application to a nozzle structure

This section presents the application of the proposed methodology to a parameterized nozzle structure. The analysis begins with a description of the geometry and physical characteristics of the nozzle. Subsequently, the training stages are detailed using the high-dimensional, reduced-order, and hyperreduced stochastic models. The neural network training strategy is outlined, followed by a comprehensive error quantification of the hyperreduced model. The computational efficiency achieved through hyperreduction is then assessed, and finally, a convergence analysis is performed, including a study of the impact of using a smaller number of realizations for training the ANN of the hyperreduced model.

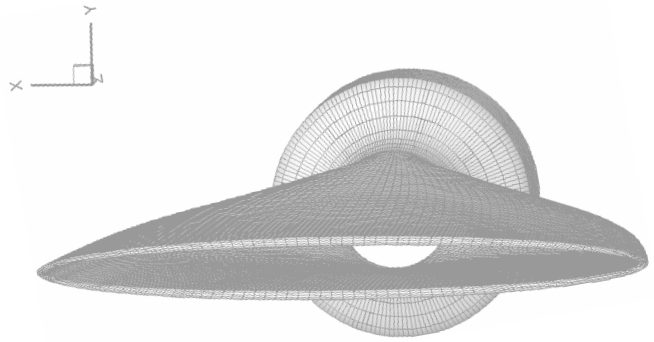

Figure 1: Finite element mesh of the nozzle structure.

6.1. Nozzle structure

As an application, an aerospace nozzle structure is considered. The FE model has $n_{\text{FE}} = 216{,}720$ DOF and the mesh is shown in Fig. 1. Except for the exit geometry and excitation forces, the structure and FE modeling are the same as in [40]. The internal surface of the nozzle is subjected to a normal pressure excitation p such that

$$p(z, \theta, t; \mathbf{w}) = w_1 \, f(z) \, g(\theta) \, h(t, w_2, w_3) \, , \tag{21}$$

in which $f(z)$ and $g(\theta)$ are given functions that describe the spatial pressure profile and where $h(t, w_2, w_3)$ is a sinc function for which the Fourier transform yields the rectangular function $\omega \mapsto 2\pi \times \mathbb{1}_{[w_2, w_3]}$ rad/s in the frequency domain. The control parameter $\mathbf{w}$ is given by $\mathbf{w} = (w_1, w_2, w_3)$, where w_1 , w_2 , and w_3 are, respectively, a dimensionless force amplitude multiplier, the excitation lower frequency bound (Hz), and the excitation upper frequency bound (Hz). The vector $\mathbf{f}^{\text{exc}}(t; \mathbf{w}, \mathbf{u})$ in Eq. (1) does not depend on $\mathbf{u}$ and we write $\mathbf{f}^{\text{exc}}(t; \mathbf{w})$. The nominal value $\underline{\mathbf{w}}$ for $\mathbf{w}$ is given by $\underline{w}_1 = 0.005$, $\underline{w}_2 = 70$ Hz, and $\underline{w}_3 = 800$ Hz. The symmetry class of the nominal elastic material is isotropic. Let $[\underline{C}]$ denote the (6×6) elasticity matrix in Voigt notation for the nominal material. The statistical fluctuations of the material with uncertainties is in the anisotropic class, for which the random elasticity matrix is denoted by $[\mathbf{C}]$. The probabilistic model of $[\mathbf{C}]$ is given in [41, 42] and is written as $[\mathbf{C}] = [\underline{L}_c]^T [\mathbf{G}(\delta)] [\underline{L}_c]$ in which $[\underline{L}_c]$ is the Cholesky factor such that $[\underline{C}] = [\underline{L}_c]^T [\underline{L}_c]$ and where $[\mathbf{G}(\delta)]$ is a positive-definite random matrix such that $E\{[\mathbf{G}(\delta)]\} = [I_6]$. The dispersion hyperparameter δ controls the amount of uncertainties and in this application, it is chosen as $\delta = 0.1$. The probabilistic model of $[\mathbf{G}(\delta)]$ is expressed as a function of a $\mathbb{R}^{n_u}$-valued random variable $\mathbf{U}$ (uncontrolled parameter) with $n_u = 21$ independent components. Since the nonlinear elastic forces $\mathbf{f}^{\text{NL}}$ with values in $\mathbb{R}^{n_{\text{FE}}}$ depend on the elastic coefficients, the vector of reduced nonlinear forces $\boldsymbol{\mathcal{F}}^{\text{NL}}$ with values in $\mathbb{R}^{n_{\text{POD}}}$ depends on $\mathbf{u}$ and the nonlinear application is written as $(\mathbf{q}, \mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q}, \mathbf{u})$ (it does not depend on $\mathbf{w}$).

6.2. Training stage using the parameterized stochastic high-dimensional model

As previously explained, the control parameter $\mathbf{w} = (w_1, w_2, w_3)$ is modeled by the random variable $\mathbf{W} = (W_1, W_2, W_3)$. It is defined as

$$W_i \sim \mathcal{U}\left[0.9\,\underline{w}_i\,, 1.1\underline{w}_i\right], \quad i = 1, 2, 3\,, \tag{22}$$

that is, the components are independent and each one follows a uniform distribution. The time simulation is performed with $N_t = 4{,}800$ time steps over the time interval $T = [0, 0.1]$s. An MC simulation with $n_{\text{train}} = 384$ realizations is performed, using the HDM. The error tolerance ε_{tol} such that $\varepsilon_{\text{POD}}(n_{\text{POD}}) \leq \varepsilon_{\text{tol}}$ is chosen as $\varepsilon_{\text{tol}} = 10^{-3}$, which yields $n_{\text{POD}} = 152$.

6.3. Repeated training stage using the parameterized stochastic reduced-order model

The MC simulation with $n_{\text{train}} = 384$ realizations is rerun with the stochastic ROM (dimension $n_{\text{POD}} = 152$) in place of the stochastic HDM. The realizations of random parameters $\mathbf{W}$ and $\mathbf{U}$ are taken as identical to those used for the initial training stage with the HDM. The ANN full training dataset $\mathcal{D}_{\text{ANN}}$, defined in Eq. (14), is collected and stored to disk. Its number of data points is $N_{\text{data}} = 4{,}521{,}049$, from which it can be deduced that $\nu_{\text{iter}} \simeq 2.45$.

6.4. Neural network training

We consider a fully connected ANN with n_{HL} hidden layers and ReLU activation functions. The ADAM optimization algorithm is used to find the weights and biases of the ANN. A key input to ADAM is the learning rate as a function of the epoch number. The learning rate function $e \mapsto \ell_r(e)$ is defined as

$$\ell_r(e) = \max(i_{\text{LR}} \times d_{\text{F}}^{\lfloor (e-1)/d_{\text{P}} \rfloor}, m_{\text{LR}})\,, \tag{23}$$

where e is the epoch number, i_{LR} is the initial learning rate, d_{F} is the decay factor, d_{P} is the decay period, and m_{LR} is the minimum learning rate. The ANN full training dataset $\mathcal{D}_{\text{ANN}}$ is split as $\mathcal{D}_{\text{ANN}} = \mathcal{D}_{\text{ANN},1} \cup \mathcal{D}_{\text{ANN},2} \cup \mathcal{D}_{\text{ANN},3}$ according to a partitioning of the $n_{\text{train}} = 384$ realizations such that $n_{\text{train},1} = 284$, $n_{\text{train},2} = 50$, and $n_{\text{train},3} = 50$. This yields $N_{\text{data},1} = 3{,}342{,}948$ data points for the ANN training subset $\mathcal{D}_{\text{ANN},1}$, $N_{\text{data},2} = 589{,}210$ data points for the ANN validation subset $\mathcal{D}_{\text{ANN},2}$, and $N_{\text{data},3} = 588{,}891$ data points for the ANN test subset $\mathcal{D}_{\text{ANN},3}$. The ANN training subset $\mathcal{D}_{\text{ANN},1}$ is then split into n_{MB} mini-batches, and the ADAM optimization is performed one mini-batch at a time. A random search is used to determine the best ANN architecture. The number n_{HL} of hidden layers satisfies $n_{\text{HL}} \in \{1, \ldots, 7\}$. The width of each hidden layer is drawn at random within ranges specified in advance, and the parameter vector $\boldsymbol{\theta}$ (weights and biases) is randomly initialized. Networks are trained while varying n_{HL}, n_{MB}, i_{LR}, d_{F}, d_{P}, m_{LR}, and the number of neurons per layer. The maximum validation accuracy $A_{\text{val}}(\boldsymbol{\theta})$ (Eq. (20)) is obtained for $n_{\text{HL}} = 1$. As a result, n_{HL} is fixed to $n_{\text{HL}} = 1$ for the remainder of the study. The remaining hyperparameters are tuned, namely n_{MB}, i_{LR}, d_{F}, d_{P}, m_{LR}, as well as the number of neurons in the single hidden layer, denoted by n_{neu}.

The ANN training is then structured as an optimization problem with two nested loops:

- an outer loop, which consists in searching the optimal values for n_{neu} , n_{MB} , i_{LR} , d_{F} , d_{P} , and m_{LR} , so as to maximize the accuracy $A_{\text{val}}(\boldsymbol{\theta})$.

- an inner loop for which, when all six above parameters are fixed, the optimal ANN weights and biases $\boldsymbol{\theta}$ are sought by ADAM optimizer for maximizing $A_{\text{val}}(\boldsymbol{\theta})$ through minimizing the cost function $J(\boldsymbol{\theta})$ defined in Eq. (19), for each mini-batch and for several epochs.

Let $\boldsymbol{\xi}$ be the vector in $\mathbb{R}^6$ such that $\boldsymbol{\xi} = (n_{\text{neu}}, n_{\text{MB}}, i_{\text{LR}}, d_{\text{F}}, d_{\text{P}}, m_{\text{LR}})$. Let $\mathcal{S} = \Pi_{j=1}^{6}[a_j, b_j]$ with $0 < a_j < b_j < +\infty$ be a given compact subset of $\mathbb{R}^6$. The vector $\boldsymbol{\xi}$ is sampled using a uniform distribution on $\mathcal{S}$ of $\boldsymbol{\xi}$. These sampled values are used for the outer loop. Note that the subset $\mathcal{S}$ is adapted during the ANN optimization. More precisely, $\mathcal{S}$ is refined over a small number of search cycles. After each cycle, scatter plots of the validation accuracy $A_{\text{val}}(\boldsymbol{\theta})$ versus each parameter are inspected. If a clear high-accuracy region emerges for the j-th parameter, interval $[a_j, b_j]$ is tightened to cover that region (the span of the best-performing points is visually selected and a small manual safety margin is added to preserve coverage) while the other bounds are left unchanged. If no clear concentration is visible for a parameter, its bounds are kept as they are. These cycles are repeated until the bounds stabilize and $A_{\text{val}}(\boldsymbol{\theta})$ no longer improves substantially. Moreover, for a given value of $\boldsymbol{\xi}$, if the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ does not increase sufficiently quickly, then the inner optimization loop is automatically stopped, according to some stop criterion (in addition to usual convergence stops). Using checkpoints indexed by p, the stop criterion is defined as follows. If the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ has not reached the value $A_{\text{val}}^{(p)}$ after the ANN has been trained for a prescribed time $\tau^{(p)}$, its training is aborted. The p-th checkpoint is defined through the pair $(\tau^{(p)}, A_{\text{val}}^{(p)})$, and the set of checkpoints is regularly updated to increase the rigor of the stop criterion. This retains only the most promising candidates and saves time. For a given ANN candidate corresponding to a given $\boldsymbol{\xi}$, the function $\tau \mapsto A_{\text{val}}(\boldsymbol{\theta}(\tau))$ is monitored, where τ denotes the elapsed time devoted to optimizing the ANN parameters, $\boldsymbol{\theta}(\tau)$, which evolve as time goes on. In particular, ensembles of functions $\tau \mapsto A_{\text{val}}(\boldsymbol{\theta}(\tau))$ corresponding to the best candidates are monitored and compared to the checkpoints $\{(\tau^{(p)}, A_{\text{val}}^{(p)})\}_p$, based on which the checkpoints are progressively tightened. Although a tight checkpoint policy saves time by discarding poor ANN candidates, it may also reject strong candidates whose accuracy improves slowly at first and peaks only after prolonged training. Throughout the training cycles, the checkpoints were manually and carefully selected and updated to save time while avoiding the rejection of the best candidates. We emphasize that both the tightening of bounds and the checkpoint schedule were adjusted manually by visual inspection, rather than by an automated procedure. This deliberate choice favors transparency and conservatism over additional algorithmic complexity.

Upon convergence of accuracy $A_{\text{val}}(\boldsymbol{\theta})$ obtained with stabilized bounds for $\mathcal{S}$, the best ANN is selected as the one with highest accuracy. The best ANN found exhibits an accuracy $A_{\text{val}}(\boldsymbol{\theta}) = 98.88\%$, which can be interpreted as an error of 1.12%. The ANN has $n_{\text{neu}} = 4{,}166$ neurons and was found upon using $n_{\text{MB}} = 49$ mini-batches and a learning rate function defined by $i_{\text{LR}} = 0.098$, $d_{\text{F}} = 0.87$, $d_{\text{P}} = 16.8$, and $m_{\text{LR}} = 4.75 \times 10^{-8}$. A total of 1,196 neural networks were tried out, among which the vast majority were not trained until convergence of their accuracy $A_{\text{val}}(\boldsymbol{\theta})$ (due to the stop criterion). Such ANN training stage lasted for about 8.7 days and was carried out using a workstation with 3 GPU nodes. It should be noted that such massive effort was carried out for research purposes, so as to make sure a nearly-optimal ANN was obtained. Similar effort was then carried out for each of the ten points of the convergence analysis presented in Section 6.8.2. This way, the convergence analysis is fair, as all the ten points have been obtained in trying to reach optimality. However, in practice, it is possible to obtain a very good ANN with much less computational effort (see the remark at the end of Section 6.7). The selected ANN was trained over 636 epochs in 2 h 10 min 10 s and as already mentioned, it reached an accuracy

$A_{\text{val}}(\boldsymbol{\theta}) = 98.88\%$, which can be interpreted as an error of 1.12%. However, it had already reached an error of 3.97% in 2 min 13 s after 9 epochs.

6.5. Repeated training stage using the parameterized stochastic hyperreduced model

The value of $A_{\text{val}}(\boldsymbol{\theta})$ does not directly indicate the quality of the ANN surrogate. To assess its quality, the MC simulation with $n_{\text{train}} = 384$ realizations is repeated, using the stochastic HROM for which the exact nonlinear application $(\mathbf{q}, \mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q}, \mathbf{u})$ is replaced by the ANN surrogate $(\mathbf{q}, \mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})$, and for which the exact Jacobian matrix $[\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q}, \mathbf{u})]$ is replaced by the ANN Jacobian matrix $[\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})]$. The realizations of random parameters $\mathbf{W}$ and $\mathbf{U}$ are taken as identical to those used for the other two training stages.

6.6. Error quantification for the parameterized stochastic hyperreduced model

It should be highlighted that the focus of this paper is on both minimizing and quantifying the "distance" between the ROM and the HROM (constructed through neural network hyperreduction) in terms of their predicted responses, but not between the ROM and the HDM.

To assess the quality of the proposed ANN surrogate, the $n_{\text{train},3} = 50$ realizations corresponding to the ANN test subset are compared: those computed by the stochastic HROM on the one hand, and those computed by the stochastic ROM on the other hand. The ROM response constitutes the reference response, and the gap with the HROM response characterizes the quality of the ANN surrogate. It should be noted that the $n_{\text{train},3} = 50$ realizations, corresponding to the ANN test subset, were completely unseen by the training procedure of the ANN. To quantify the HROM accuracy, $n_{\text{obs}} = 168$ normal displacements at the nozzle ellipsoidal exit are considered as observations. Then, the following quadratic error measure $\epsilon_{ij}^{\text{time}}$ is introduced to quantify the accuracy of a given deterministic trajectory (corresponding to the i-th observation and the j-th realization),

$$\epsilon_{ij}^{\text{time}} = \sqrt{\frac{\dfrac{1}{|T|}\displaystyle\int_T \left(y_{ij}^{\text{HROM}}(t) - y_{ij}^{\text{ROM}}(t)\right)^2 \mathrm{dt}}{\dfrac{1}{n_{\text{obs}}}\displaystyle\sum_{k=1}^{n_{\text{obs}}} \dfrac{1}{n_{\text{train},3}} \sum_{\ell=1}^{n_{\text{train},3}} \dfrac{1}{|T|}\displaystyle\int_T \left(y_{k\ell}^{\text{ROM}}(t)\right)^2 \mathrm{dt}}}\,, \tag{24}$$

in which $|T| = t_f - t_i$ is the length of the time interval of analysis, and where y_{ij}^{ROM} (resp. y_{ij}^{HROM}) is the time response (displacement) for observation $i \in \{1, \ldots, n_{\text{obs}}\}$ and realization $j \in \{1, \ldots, n_{\text{train},3}\}$ computed by the ROM (resp. by the HROM). Then, the global quadratic error ϵ^{time} is defined by

$$\epsilon^{\text{time}} = \sqrt{\frac{1}{n_{\text{obs}}}\sum_{i=1}^{n_{\text{obs}}} \frac{1}{n_{\text{train},3}} \sum_{j=1}^{n_{\text{train},3}} \left(\epsilon_{ij}^{\text{time}}\right)^2}\,. \tag{25}$$

It is also interesting to quantify the error related to the random response predicted by the stochastic HROM with respect to that predicted by the stochastic ROM. To do so, the overlap coefficient OVL is considered, such that

$$\mathrm{OVL}(X, Y) = 1 - \frac{1}{2}\int_{\mathbb{R}} |p_X(x) - p_Y(x)| \mathrm{dx}\,, \tag{26}$$

in which X and Y are random variables with values in $\mathbb{R}$, with p_X and p_Y their PDF, and where $|\cdot|$ denotes the absolute value. Let Y_i^{ROM} (resp. Y_i^{HROM}) denote the random variable with values

in $\mathbb{R}$ for which $\{y_{ij}^{\text{ROM}}\}_j$ (resp. $\{y_{ij}^{\text{HROM}}\}_j$) are the realizations. The overlap function $t \mapsto \text{OVL}_i^{\text{time}}(t)$ associated with observation $i \in \{1, \dots, n_{\text{obs}}\}$ is then defined as

$$\text{OVL}_i^{\text{time}}(t) = \text{OVL}\left(Y_i^{\text{HROM}}(t), Y_i^{\text{ROM}}(t)\right) . \tag{27}$$

The mean overlap $\overline{\text{OVL}}_i^{\text{time}}$ associated with observation i is then such that

$$\overline{\text{OVL}}_i^{\text{time}} = \frac{1}{|T|} \int_T \text{OVL}_i^{\text{time}}(t) \text{dt} . \tag{28}$$

At last, the global overlap OVL^{time} is defined as

$$\text{OVL}^{\text{time}} = \frac{1}{n_{\text{obs}}} \sum_{i=1}^{n_{\text{obs}}} \overline{\text{OVL}}_i^{\text{time}} . \tag{29}$$

We are also interested in evaluating frequency response functions (FRF) in the frequency band of analysis $\mathcal{B} = [\omega_{\min}, \omega_{\max}]$ with $\omega_{\min} = 2\pi \times 40$ rad/s and $\omega_{\max} = 2\pi \times 1300$ rad/s (between 40 Hz and 1300 Hz). The same error measures (global quadratic error and global overlap) are considered for the FRF. Let $\widehat{y}_{ij}^{\text{ROM}}$ (resp. $\widehat{y}_{ij}^{\text{HROM}}$) denote the logarithm of the modulus of the frequency response (acceleration) for observation $i \in \{1, \dots, n_{\text{obs}}\}$ and realization $j \in \{1, \dots, n_{\text{train},3}\}$ computed by the ROM (resp. by the HROM) through Fourier transform of y_{ij}^{ROM} (resp. y_{ij}^{HROM}). Similarly to Eq. (24), the quadratic error measure $\epsilon_{ij}^{\text{freq}}$ is then introduced, such that

$$\epsilon_{ij}^{\text{freq}} = \sqrt{\frac{\frac{1}{|\mathcal{B}|} \int_{\mathcal{B}} \left(\widehat{y}_{ij}^{\text{HROM}}(\omega) - \widehat{y}_{ij}^{\text{ROM}}(\omega)\right)^2 \text{d}\omega}{\frac{1}{n_{\text{obs}}} \sum_{k=1}^{n_{\text{obs}}} \frac{1}{n_{\text{train},3}} \sum_{\ell=1}^{n_{\text{train},3}} \frac{1}{|\mathcal{B}|} \int_{\mathcal{B}} \left(\widehat{y}_{k\ell}^{\text{ROM}}(\omega)\right)^2 \text{d}\omega}} , \tag{30}$$

in which $|\mathcal{B}| = \omega_{\max} - \omega_{\min}$ is the length of the frequency band of analysis, and where ω denotes the circular frequency (rad/s). The global quadratic error ϵ^{freq} is then defined as

$$\epsilon^{\text{freq}} = \sqrt{\frac{1}{n_{\text{obs}}} \sum_{i=1}^{n_{\text{obs}}} \frac{1}{n_{\text{train},3}} \sum_{j=1}^{n_{\text{train},3}} \left(\epsilon_{ij}^{\text{freq}}\right)^2} . \tag{31}$$

Let $\widehat{Y}_i^{\text{ROM}}$ (resp. $\widehat{Y}_i^{\text{HROM}}$) denote the random variable with values in $\mathbb{R}$ for which $\{\widehat{y}_{ij}^{\text{ROM}}\}_j$ (resp. $\{\widehat{y}_{ij}^{\text{HROM}}\}_j$) are the realizations. Similarly to Eq. (27), the overlap function $\omega \mapsto \text{OVL}_i^{\text{freq}}(\omega)$ associated with observation $i \in \{1, \dots, n_{\text{obs}}\}$ is then defined as

$$\text{OVL}_i^{\text{freq}}(\omega) = \text{OVL}\left(\widehat{Y}_i^{\text{HROM}}(\omega), \widehat{Y}_i^{\text{ROM}}(\omega)\right) . \tag{32}$$

The mean overlap $\overline{\text{OVL}}_i^{\text{freq}}$ associated with observation i is then such that

$$\overline{\text{OVL}}_i^{\text{freq}} = \frac{1}{|\mathcal{B}|} \int_{\mathcal{B}} \text{OVL}_i^{\text{freq}}(\omega) \text{d}\omega . \tag{33}$$

At last, the global overlap $\mathrm{OVL}^{\mathrm{freq}}$ is defined as

$$\mathrm{OVL}^{\mathrm{freq}} = \frac{1}{n_{\mathrm{obs}}} \sum_{i=1}^{n_{\mathrm{obs}}} \overline{\mathrm{OVL}}_i^{\mathrm{freq}}. \tag{34}$$

The results are as follows. The global quadratic errors in time and frequency are $\epsilon^{\mathrm{time}} = 1.66\%$ and $\epsilon^{\mathrm{freq}} = 2.00\%$, while the global overlaps in time and frequency are $\mathrm{OVL}^{\mathrm{time}} = 99.0\%$ and $\mathrm{OVL}^{\mathrm{freq}} = 97.4\%$, from Gaussian kernel density estimation (GKDE) using the Silverman bandwidth. The following figures compare the predictions of the proposed HROM to those of the ROM. The predictions of the ROM serve as a reference. In all the figures, the predicted responses by the HROM are depicted in blue, whereas those predicted by the ROM are depicted in black. Two types of random responses are presented, namely the 96% confidence region, and the PDF of the response at a given time or frequency. The 96% confidence regions are obtained by discarding, for each time step (or frequency point), the largest and smallest responses among the $n_{\mathrm{train},3} = 50$ realizations. Furthermore, the PDF of a response is estimated using GKDE (with the Silverman bandwidth), and so are the resulting overlap (OVL) values.

Figure 2 displays the time response for a given realization and for a given observation, for each model (the ROM and the HROM), and for which the quadratic error is similar to the global quadratic error in time. It can be seen that the two curves are almost perfectly superimposed, which demonstrates the very good accuracy of the proposed neural network hyperreduction approach for this application. Figure 3 depicts the estimated random time response (96% confidence region) for a given observation, for each model, and for which the mean overlap is similar to the global overlap. Once again, both responses are almost perfectly superimposed. Next, Figure 4 shows the estimated PDF at a given time of the same observation as Figure 3, for each model, and for which the overlap is similar to the global overlap. A very good agreement can be observed. Finally, Figures 5 to 7 present the same comparisons as Figures 2 to 4, but for the frequency response. The conclusion is similar, that is the HROM yields predictions very close to those of the ROM. Nonetheless, the gap is slightly more significant, especially in higher frequencies. Two reasons may explain the increased error observed in the frequency domain. First, differences in low-amplitude components are exacerbated by the logarithmic scale. Second and more importantly, since the overlap tends to be higher at low frequencies, the choice of the upper frequency bound of analysis can significantly affect the global overlap.

Remark on the number of realizations for PDF estimation. With $n_{\mathrm{train},3} = 50$ realizations, statistical convergence of the random responses is not attained (see Figs. C.36–C.37 in Appendix C). Nevertheless, a meaningful quantification of the ROM–HROM discrepancy is obtained by comparing random responses generated with identical draws of the random variables, which provides a perspective complementary to the trajectory-level agreement in Figs. 2 and 5. The test set of size $n_{\mathrm{train},3} = 50$ is a subset of the $n_{\mathrm{train}} = 384$ available draws. The value $n_{\mathrm{train}} = 384$ was chosen to provide sufficient data and diversity to train the ANN. It was not chosen to make a typical test subset proportion between 10% and 20% in a three way split, that is $n_{\mathrm{train},3} = 50$, large enough to ensure Monte Carlo convergence of the PDF.

Overall, it can be concluded that the proposed HROM, relying on an ANN surrogate trained on $n_r = n_{\mathrm{train},1} + n_{\mathrm{train},2} = 284 + 50 = 334$ realizations, is very accurate. It should be noted that

this excellent accuracy of the HROM is obtained despite the presence of very strong nonlinear effects, see Appendix B.

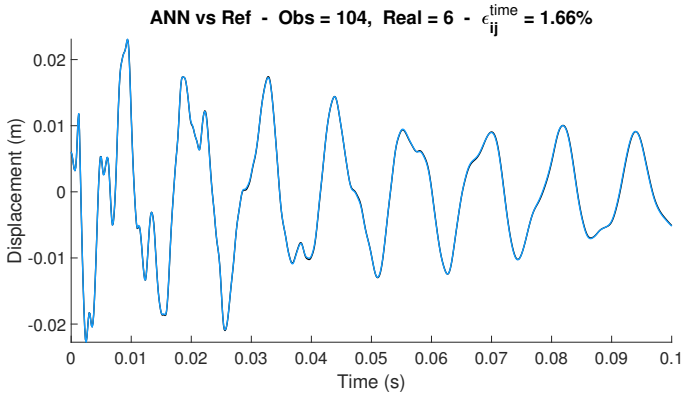

Figure 2: Time response for observation $i = 104$ and realization $j = 6$ (among $n_{\text{train},3} = 50$), for which $\epsilon_{ij}^{\text{time}} \simeq \epsilon^{\text{time}} \simeq 0.0166$. Reference ROM (black) and ANN HROM (blue).

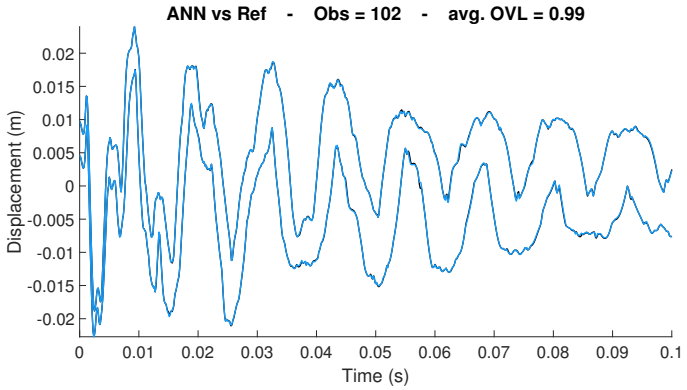

Figure 3: Random time response (96% confidence region over $n_{\text{train},3} = 50$ realizations), for observation $i = 102$, for which $\overline{\text{OVL}}_i^{\text{time}} \simeq \text{OVL}^{\text{time}} \simeq 0.99$. Reference ROM (black) and ANN HROM (blue).

6.7. Computational savings

This section compares the runtime of the HROM to that of the ROM. It does not compare to the HDM, because ANN hyperreduction is designed to accelerate the ROM, irrespective of any ROM–HDM speed-up. For completeness, during data generation (the two training stages), $n_{\text{train}} = 384$ realizations were run in parallel across 96 workers on a 96-core workstation, once for each training stage (the first one with the HDM and the second one with the ROM). The wall-clock time was 25.8 days for the HDM and 4.8 days for the ROM, therefore exhibiting a 5.3× speed-up of the ROM over the HDM. This supports our choice of *Option 2* over *Option 1* in Section 5.1.

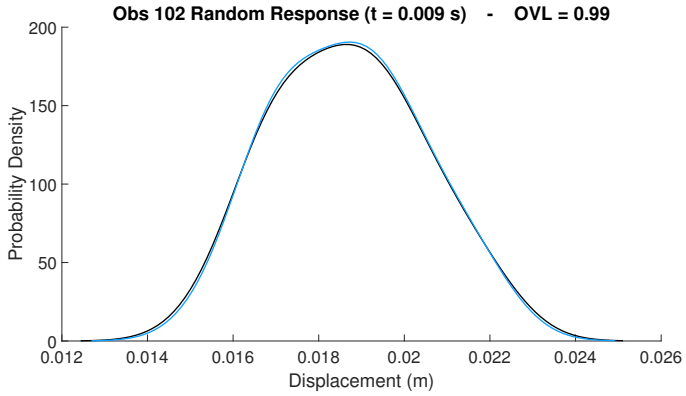

Figure 4: Time response PDF (GKDE with $n_{\text{train},3} = 50$ realizations), at time $t_{ex} = 0.009$ s and for observation $i = 102$, for which $\text{OVL}_i^{\text{time}}(t_{ex}) \simeq \text{OVL}^{\text{time}} \simeq 0.99$. Reference ROM (black) and ANN HROM (blue).

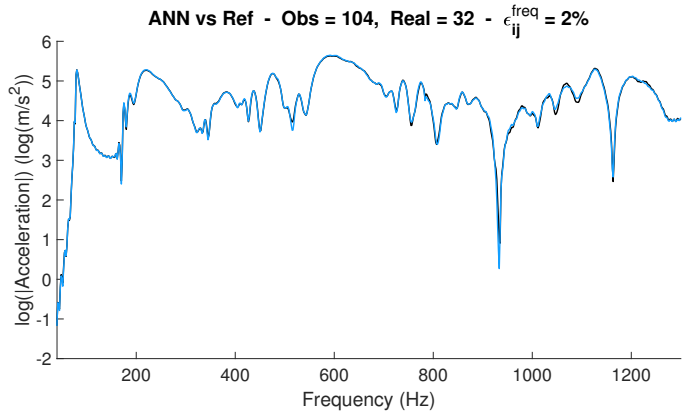

Figure 5: Frequency response for observation $i = 104$ and realization $j = 32$ (among $n_{\text{train},3} = 50$), for which $\epsilon_{ij}^{\text{freq}} \simeq \epsilon^{\text{freq}} \simeq 0.0200$. Reference ROM (black) and ANN HROM (blue).

Now, the computational gain enabled by the HROM, in comparison to the ROM, and related to replacing the exact nonlinear forces with the ANN surrogate and the Jacobian matrices with the ANN derivatives, is as follows. The ROM was run on a workstation with 96 CPU cores, whereas the HROM was run on a workstation with 112 CPU cores and 3 GPU nodes. In both cases, all machine resources were fully exploited (full charge). The GPU nodes were used to efficiently perform the forward evaluations of the ANN as well as computing its derivatives by backpropagation. For both models, the $n_{\text{train}} = 384$ realizations were executed in parallel, over 96 workers for the ROM and 112 workers for the HROM.

On average over the $n_{\text{train},3} = 50$ realizations corresponding to the test subset $\mathcal{D}_{\text{ANN},3}$, the elapsed time to evaluate $\boldsymbol{\mathcal{F}}^{\text{NL}}$ with one worker, was found to be about 8 seconds. In comparison, to evaluate $\boldsymbol{\mathcal{F}}^{\text{ANN}}$ with the ANN, it was found to be about 2 milliseconds. Next, regarding the elapsed times devoted to construct the Jacobian matrices $[\text{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q},\mathbf{u})]$ and $[\text{D}_{\mathbf{q}}\boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q},\mathbf{u})]$ throughout the time simulations, they were found to be about 35 seconds for the ROM and about 0.7 seconds

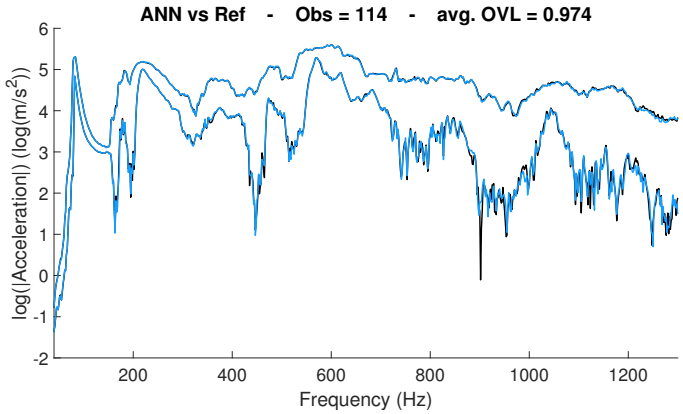

Figure 6: Random frequency response (96% confidence region over $n_{\text{train},3} = 50$ realizations), for observation $i = 114$, for which $\overline{\text{OVL}}_i^{\text{freq}} \simeq \text{OVL}^{\text{freq}} \simeq 0.974$. Reference ROM (black) and ANN HROM (blue).

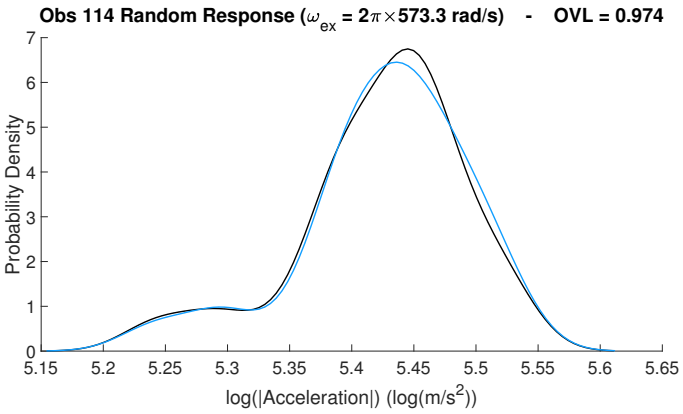

Figure 7: Frequency response PDF (GKDE with $n_{\text{train},3} = 50$ realizations), at frequency $\omega_{ex} = 2\pi \times 573.3$ rad/s and for observation $i = 114$, for which $\text{OVL}_i^{\text{freq}}(\omega_{ex}) \simeq \text{OVL}^{\text{freq}} \simeq 0.974$. Reference ROM (black) and ANN HROM (blue).

for the HROM (for the latter, through efficient neural network backpropagation using the GPU nodes). Last, regarding the total elapsed times, the $n_{\text{train},3} = 50$ realizations were performed in about 29 hours using the ROM and in about 4 minutes using the HROM, which constituted a speed-up factor of about 400.

Remarks on the offline computational cost. The offline costs of the HDM/ROM simulations and ANN training are now quantified. The 384 ROM realizations required 4.8 days and training the ANN required 8.7 days. The same wall time corresponds to about 201 HDM realizations, and the 8.7 days amount to about 696 ROM realizations. Such numbers of realizations are very small relative to the online capability of the HROM, which computed 24,640 realizations in about 30 hours (performed three times for the analyses in Appendix C). Importantly, part of the aforementioned offline computational cost reflects exploratory research. A very accurate HROM can be obtained with a fraction of this cost. For example, training with 53 realizations instead

of 334 yields a validation error of 4% rather than 1.12%, while the resulting trajectories still exhibit excellent pointwise agreement. For reference, with 334 realizations the validation error already reaches 3.97% after only 9 epochs in 2 min 13 s before converging to 1.12%. Since the primary focus is the HROM speed-up relative to the ROM, which enables tens of thousands of realizations, the offline cost is not analyzed further.

6.8. Convergence analysis

6.8.1. Hyperreduction with smaller number of realizations

The proposed HROM for which an excellent accuracy was shown in Section 6.6, relied on an ANN trained on $n_r = n_{\text{train},1} + n_{\text{train},2} = 334$ realizations. In this section, the effectiveness of the proposed neural network hyperreduction approach is investigated with respect to the number n_r of available expensive simulations that are necessary to train the ANN. To this end, the error quantification for the HROM is presented for the new case $n_r = n_{\text{train},1} + n_{\text{train},2} = 45 + 8 = 53$.

Thus, the ANN is trained as previously described, but this time on a significantly smaller dataset. Upon convergence of the accuracy $A_{\text{val}}(\boldsymbol{\theta})$ obtained with stabilized bounds for $\mathcal{S}$, the best ANN is selected as the one with highest accuracy. The best ANN found exhibits an accuracy $A_{\text{val}}(\boldsymbol{\theta}) = 95.97\%$, which can be interpreted as an error of 4.03%. The ANN has $n_{\text{neu}} = 665$ neurons and was found upon using $n_{\text{MB}} = 47$ mini-batches and a learning rate function defined by $i_{\text{LR}} = 0.054$, $d_{\text{F}} = 0.967$, $d_{\text{P}} = 12.7$, and $m_{\text{LR}} = 3.73 \times 10^{-7}$. A total of 11,981 neural networks were tried out, among which the vast majority were stopped before reaching convergence. Such ANN training stage lasted for about 4.1 days. It should be noted that in practice, it is possible to obtain a very good ANN with much less computational effort. The ANN was trained over 1,208 epochs in 26 min 8 s and as already mentioned, it reached an accuracy $A_{\text{val}}(\boldsymbol{\theta}) = 95.97\%$, which can be interpreted as an error of 4.03%. However, it had already reached an error of 6.00% in 1 min 13 s after 50 epochs.

The results are as follows. The global quadratic errors in time and frequency are $\epsilon^{\text{time}} = 6.44\%$ and $\epsilon^{\text{freq}} = 5.96\%$, while the global overlaps in time and frequency are $\text{OVL}^{\text{time}} = 95.7\%$ and $\text{OVL}^{\text{freq}} = 92.0\%$. The next six figures (Figures 8 to 13) mirror the six figures presented in previous section (Figures 2 to 7). The structure and content of each figure correspond exactly, and thus no further description is provided. Figures 8 to 13 demonstrate a relatively good accuracy of the proposed HROM trained on only $n_r = 53$ expensive realizations. This tends to show that the proposed ANN hyperreduction approach still remains valid for the case of relatively small data.

6.8.2. Convergence of the hyperreduction with respect to number of realizations

The results have been presented for $n_r = 334$ and then for $n_r = 53$. The work has also been carried out for eight additional values of n_r, so as to perform a convergence analysis with respect to n_r. For this convergence analysis, the ensemble of n_r values is such that $n_r \in \{27, 33, 40, 53, 67, 87, 120, 167, 240, 334\}$. Figure 14 shows the number n_{neu} of neurons as a function of the number n_r of expensive realizations used for training the ANN. Qualitatively, the relationship is linear, meaning the more the training data, the bigger the ANN. It also means that a small training dataset yields a small ANN. It should be noted that for the case of small data, a bigger ANN (more neurons) could allow the loss function J defined in Eq. (19) to improve, whereas the resulting ANN accuracy $A_{\text{val}}(\boldsymbol{\theta})$ would not. This is explained by the fact that the

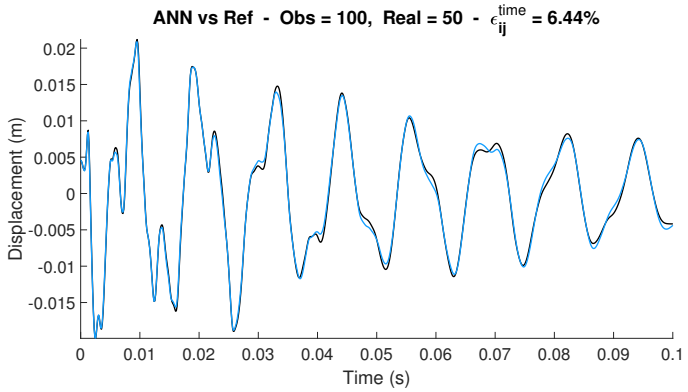

Figure 8: Time response for observation $i = 100$ and realization $j = 50$ (among $n_{\text{train},3} = 50$), for which $\epsilon_{ij}^{\text{time}} \simeq \epsilon^{\text{time}} \simeq 0.0644$. Reference ROM (black) and ANN HROM (blue).

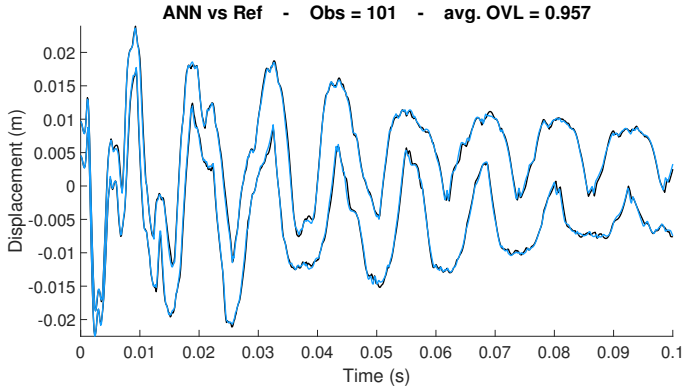

Figure 9: Random time response (96% confidence region over $n_{\text{train},3} = 50$ realizations), for observation $i = 101$, for which $\overline{\text{OVL}}_i^{\text{time}} \simeq \text{OVL}^{\text{time}} \simeq 0.957$. Reference ROM (black) and ANN HROM (blue).

ANN accuracy is defined upon the ANN validation subset $\mathcal{D}_{\text{ANN},2}$, as opposed to the ANN training subset $\mathcal{D}_{\text{ANN},1}$. Thus, such a choice of ANN accuracy as given in Eq. (20) prevents overfitting, by construction. Figure 15 shows the ANN accuracy $A_{\text{val}}(\boldsymbol{\theta})$ as a function of n_r. It can be observed that the ANN accuracy $A_{\text{val}}(\boldsymbol{\theta})$ quickly increases toward a high value above 95% (as soon as $n_r = 40$) and then slowly increases to nearly reach 99% for the maximum value considered ($n_r = 334$). Next, Figure 16 shows the total number of Newton-Raphson iterations with respect to n_r and it can be seen that a small training dataset (with n_r below 50) is accompanied with an increased number of iterations of the nonlinear solver. Nonetheless, the number of such iterations remains relatively stable. In addition, Figure 17 depicts the total number of Jacobian matrix updates as a function of n_r. It should be mentioned that for efficiency concerns, the Jacobian matrix was not updated at every iteration. Instead, it was only updated according to some criteria defined upon the history of the time simulation (see Appendix D). Such criteria involved, for instance, the number of nonlinear iterations on both the current and previous time steps. This figure shows the same behavior as previous figure. Namely, both the number of nonlinear iterations and Jacobian matrix updates increase as the training dataset size decreases (i.e., when

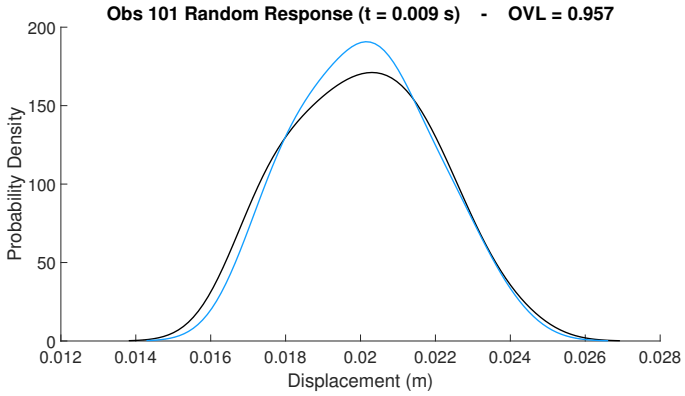

Figure 10: Time response PDF (GKDE with $n_{\text{train},3} = 50$ realizations), at time $t_{ex} = 0.009$ s and for observation $i = 101$, for which $\text{OVL}_i^{\text{time}}(t_{ex}) \simeq \text{OVL}^{\text{time}} \simeq 0.957$. Reference ROM (black) and ANN HROM (blue).

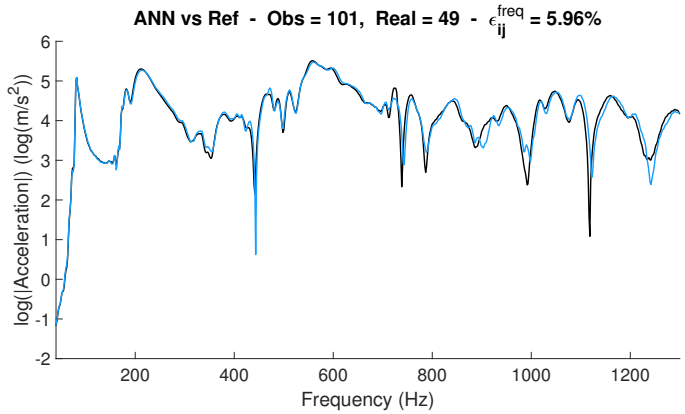

Figure 11: Frequency response for observation $i = 101$ and realization $j = 49$ (among $n_{\text{train},3} = 50$), for which $\epsilon_{ij}^{\text{freq}} \simeq \epsilon^{\text{freq}} \simeq 0.0596$. Reference ROM (black) and ANN HROM (blue).

n_r decreases). The next and last four Figures 18 to 21 present the four global error measures (quadratic error and overlap, both in time and in frequency) as a function of n_r . The quadratic error in time quickly decreases below 5% and then slowly decreases below 2%. Similarly, the quadratic error in frequency quickly decreases below 6% and then slowly decreases to nearly reach 2%. Conversely, the overlap in time quickly increases above 95% and then slowly increases to nearly reach 99%, and the overlap in frequency quickly increases above 90% and then slowly increases to reach 97%. For all these convergence curves, the convergence is relatively smooth and stable with respect to n_r , which demonstrates the robustness and reliability of the proposed neural network hyperreduction method.

A potential objection could be that the good performance observed is merely due to the rapid convergence of the random response with respect to the number of MC realizations — implying that the training database is already large enough and that no additional realizations are required. This would question the usefulness of the HROM, whose purpose is to generate further real-

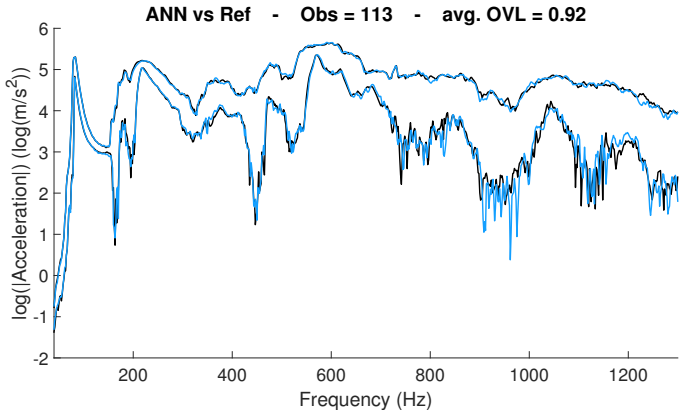

Figure 12: Random frequency response (96% confidence region over $n_{\text{train},3} = 50$ realizations), for observation $i = 113$, for which $\overline{\text{OVL}}_i^{\text{freq}} \simeq \text{OVL}^{\text{freq}} \simeq 0.92$. Reference ROM (black) and ANN HROM (blue).

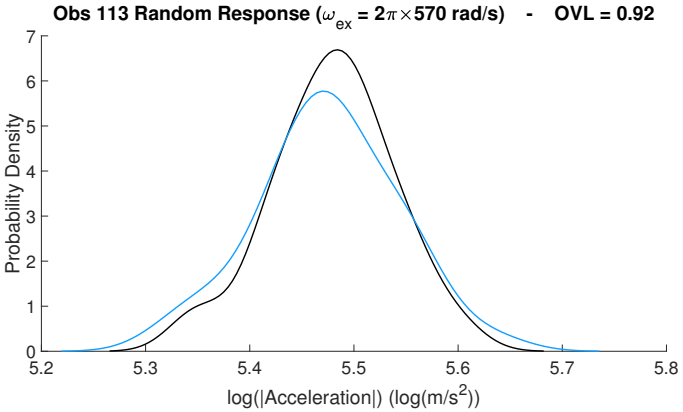

Figure 13: Frequency response PDF (GKDE with $n_{\text{train},3} = 50$ realizations), at frequency $\omega_{ex} = 2\pi \times 570$ rad/s and for observation $i = 113$, for which $\text{OVL}_i^{\text{freq}}(\omega_{ex}) \simeq \text{OVL}^{\text{freq}} \simeq 0.92$. Reference ROM (black) and ANN HROM (blue).

izations at a very reduced cost. To counter this claim and further support the relevance of the proposed approach, Appendix C shows that the statistical convergence of the random response is, in fact, very slow. In particular, it focuses on the conditional random response for a fixed control parameter $\mathbf{W} = \mathbf{w}_0$, and demonstrates that the nonparametric conditional estimation converges very slowly. In contrast, the random response obtained directly using the proposed HROM with fixed $\mathbf{W} = \mathbf{w}_0$ achieves high accuracy using only a moderate number ($n_r = 53$) of expensive realizations to train the neural network.

7. Conclusions

A novel hyperreduction method using neural networks has been presented to accelerate the nonlinear simulations in stochastic dynamics. Embedded within a parameterized reduced-order modeling framework, the proposed approach takes advantage of the expensive offline phase, required

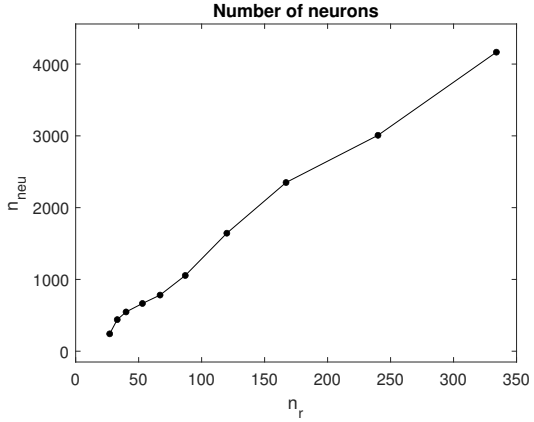

Figure 14: ANN number n_{neu} of neurons as a function of n_r.

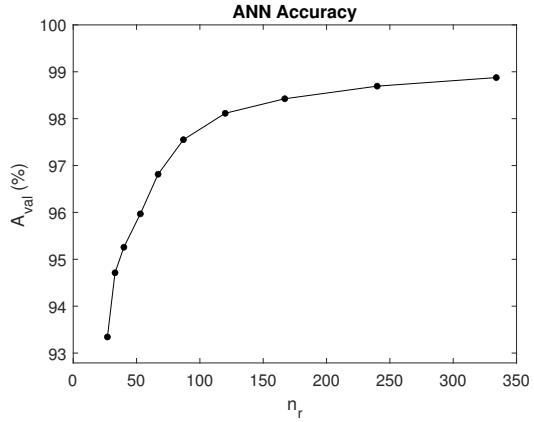

Figure 15: ANN accuracy $A_{\text{val}}(\boldsymbol{\theta})$ (%) as a function of n_r.

by the reduced-order model itself, to collect data for training the artificial neural network.

The proposed hyperreduction method relies on using such artificial neural network to replace the costly evaluation of the projected nonlinear forces and their derivatives with respect to the reduced displacements. The numerical application to an aerospace nozzle presented demonstrates the very good accuracy, efficiency, and robustness of the proposed neural network surrogate model. The proposed hyperreduced model allows for accelerating the computation by two orders of magnitude, as compared to the reduced-order model without hyperreduction. Note that the method stays efficient even for a relatively small number of realizations of the expensive computational model. The proposed machine learning approach differs from approaches that replace the entire input-output black-box computer code with an artificial neural network. Instead, it focuses on replacing the most computationally-intensive internal part of the computer code, allowing to keep most of the governing physics intact. An interesting extension of this work would be the implementation of a nonparametric probabilistic treatment of model-form uncertainties using the proposed hyperreduced model.

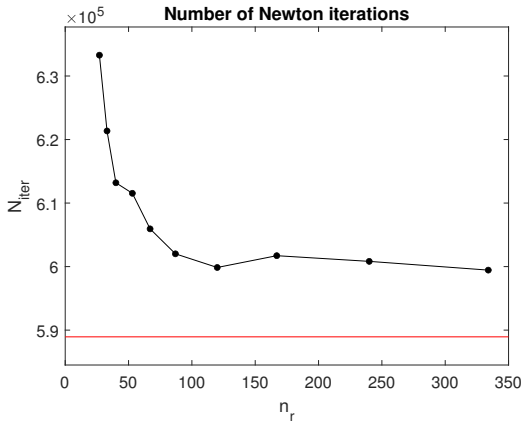

Figure 16: Total number N_{iter} of Newton-Raphson iterations as a function of n_r, for the reference ROM (red) and for the ANN HROM (black).

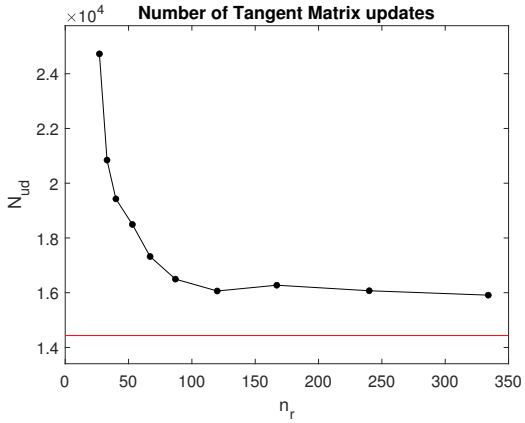

Figure 17: Total number N_{ud} of Jacobian matrix updates as a function of n_r, for the reference ROM (red) and for the ANN HROM (black).

Appendix A. Training data preprocessing

This appendix is devoted to the statistical preprocessing of the ANN training data, that was performed for the nozzle application presented in Section 6. Since, for the nozzle case study, the nonlinear application simplifies as $(\mathbf{q}, \mathbf{u}) \mapsto \boldsymbol{\mathcal{F}}^{\text{NL}}(\mathbf{q}, \mathbf{u})$, the ANN full training dataset $\mathcal{D}_{\text{ANN}}$ is rewritten as

$$\mathcal{D}_{\text{ANN}} = \{\boldsymbol{\mathcal{F}}^{\text{NL},i}, \mathbf{q}^i, \mathbf{u}^i\}_{i=1,\dots,N_{\text{data}}}, \tag{A.1}$$

in which the vectors $\mathbf{q}^i \in \mathbb{R}^{n_{\text{POD}}}$ and $\mathbf{u}^i \in \mathbb{R}^{n_u}$ correspond to the inputs of the nonlinear application whose output is $\boldsymbol{\mathcal{F}}^{\text{NL},i} \in \mathbb{R}^{n_{\text{POD}}}$, with $n_{\text{POD}} = 152$ and $n_u = 21$. That is, the nonlinear application is from $\mathbb{R}^{n_{\text{POD}}+n_u}$ to $\mathbb{R}^{n_{\text{POD}}}$, that is, from $\mathbb{R}^{173}$ to $\mathbb{R}^{152}$.

In order to normalize the ANN full training dataset and decorrelate its components, we construct a PCA representation for $\mathbf{q}$ and $\mathbf{u}$. If statistical reduction is possible, it is applied. Even without dimensionality reduction, PCA provides uncorrelated variables with unit variance, improving the

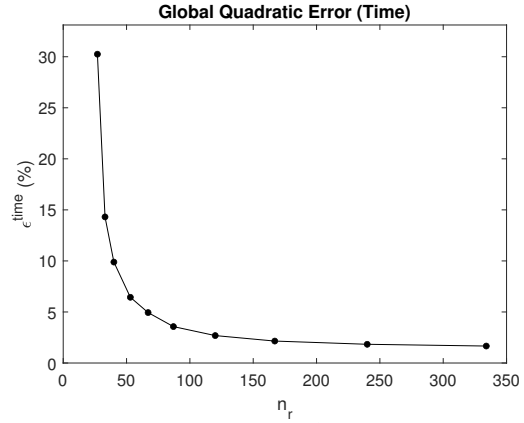

Figure 18: Global quadratic error ϵ^{time} (%) as a function of n_r.

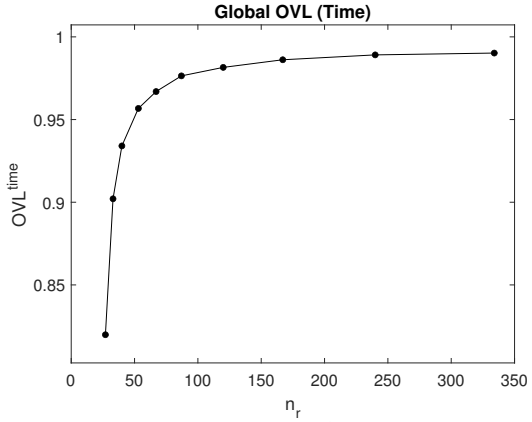

Figure 19: Global overlap OVL^{time} as a function of n_r.

conditioning of the training data.

Concerning $\mathbf{q}$, let $[q^1]$ denote the $(n_{\text{POD}} \times N_{\text{data},1})$ real matrix whose columns are the displacements $\{\mathbf{q}^i\}_i$ belonging to the ANN training subset $\mathcal{D}_{\text{ANN},1}$ with $N_{\text{data},1} = 3{,}342{,}948$ data points corresponding to $n_{\text{train},1} = 284$ realizations (see Section 5.3). The SVD of the centered displacements is considered,

$$[U_q][\Sigma_q][V_q]^T = \frac{1}{\sqrt{N_{\text{data},1} - 1}} \left([\, q^1 \,] - [\, \overline{q^1} \,] \right), \tag{A.2}$$

in which each column of matrix $[\, \overline{q^1} \,]$ is the average displacement and where $[U_q]$ is the matrix of the left-singular vectors, $[V_q]$ is the matrix of the right-singular vectors, and $[\Sigma_q]$ is the diagonal matrix of the singular values. Such SVD yields a PCA of the displacements. Let $\varepsilon^q_{\text{PCA}}$ be the PCA error such that

$$\varepsilon^q_{\text{PCA}}(n^q_{\text{PCA}}) = 1 - \frac{\sum_{k=1}^{n^q_{\text{PCA}}} \left(\sigma^q_k\right)^2}{\sum_{k=1}^{N_{\text{data},1}} \left(\sigma^q_k\right)^2}, \tag{A.3}$$

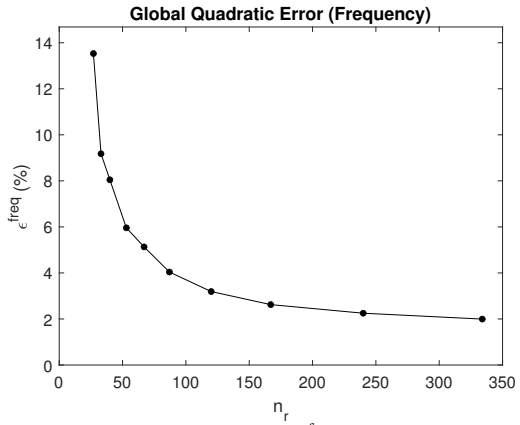

Figure 20: Global quadratic error ϵ^{freq} (%) as a function of n_r.

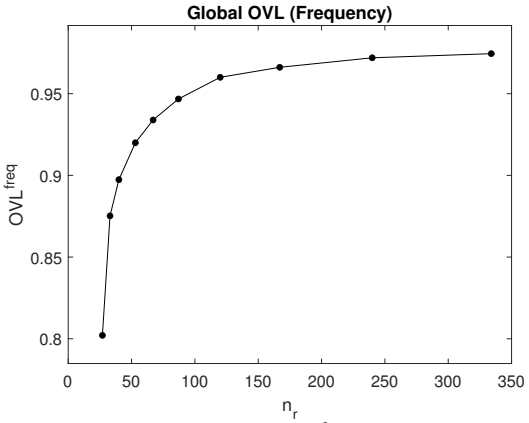

Figure 21: Global overlap OVL^{freq} as a function of n_r.

in which σ_k^q is the k-th largest singular value and where the term $\sum_{k=1}^{N_{\text{data},1}} \left(\sigma_k^q\right)^2$ is evaluated by $\frac{1}{N_{\text{data},1}} \sum_i \sum_j ([\, q^1\,] - [\,\overline{q^1}\,])_{ij}^2$. Reduced dimension $n_{\text{PCA}}^q \leq \min(N_{\text{data},1}, n_{\text{POD}})$ is the minimum number n_{PCA}^q such that $\varepsilon_{\text{PCA}}^q(n_{\text{PCA}}^q) \leq \varepsilon_{\text{tol}}^q$, with $\varepsilon_{\text{tol}}^q$ a given error tolerance. In the nozzle application of Section 6, the value $\varepsilon_{\text{tol}}^q = 10^{-6}$ is considered, which yields $n_{\text{PCA}}^q = 55$.

Likewise, the PCA of parameter $\mathbf{u}$ is also considered. Let $[u^1]$ denote the $(n_u \times N_{\text{data},1})$ real matrix whose columns are the vectors $\{\mathbf{u}^i\}_i$ belonging to the ANN training subset $\mathcal{D}_{\text{ANN},1}$. Similarly, the SVD of the centered parameters is considered, such that

$$[U_u][\Sigma_u][V_u]^T = \frac{1}{\sqrt{N_{\text{data},1} - 1}} \left([\, u^1\,] - [\,\overline{u^1}\,]\right), \tag{A.4}$$

in which each column of matrix $[\,\overline{u^1}\,]$ is the average parameter and where $[U_u]$ is the matrix of the left-singular vectors, $[V_u]$ is the matrix of the right-singular vectors, and $[\Sigma_u]$ is the diagonal

matrix of the singular values. Let $\varepsilon^u_{\text{PCA}}$ be the PCA error such that

$$\varepsilon^u_{\text{PCA}}(n^u_{\text{PCA}}) = 1 - \frac{\sum_{k=1}^{n^u_{\text{PCA}}} \left(\sigma^u_k\right)^2}{\sum_{k=1}^{N_{\text{data},1}} \left(\sigma^u_k\right)^2}\,, \tag{A.5}$$

in which σ^u_k is the k-th largest singular value and where the term $\sum_{k=1}^{N_{\text{data},1}} \left(\sigma^u_k\right)^2$ is evaluated by $\frac{1}{N_{\text{data},1}} \sum_i \sum_j ([\,u^1\,] - [\,\overline{u^1}\,])^2_{ij}$. Reduced dimension $n^u_{\text{PCA}} \leq \min(N_{\text{data},1}, n_u)$ is the minimum number n^u_{PCA} such that $\varepsilon^u_{\text{PCA}}(n^u_{\text{PCA}}) \leq \varepsilon^u_{\text{tol}}$, with $\varepsilon^u_{\text{tol}}$ a given error tolerance. In the nozzle application of Section 6, the value $\varepsilon^u_{\text{tol}} = 10^{-6}$ is considered, which yields $n^u_{\text{PCA}} = 21$ (no reduction). Such lack of statistical reduction is explained by the fact that the $n_u = 21$ components of random uncontrolled parameter $\mathbf{U}$ are statistically independent as per their construction given in Section 6.1.

For both input parameters $\mathbf{q}$ and $\mathbf{u}$, the PCA was performed on the ANN training subset $\mathcal{D}_{\text{ANN},1}$ corresponding to $n_{\text{train},1} = 284$ realizations. The ANN full training dataset $\mathcal{D}_{\text{ANN}}$ (corresponding to all the $n_{\text{train}} = 384$ realizations) is then substituted with the modified dataset $\widetilde{\mathcal{D}}_{\text{ANN}}$ such that

$$\widetilde{\mathcal{D}}_{\text{ANN}} = \{\boldsymbol{\mathcal{F}}^{\text{NL},i}, \boldsymbol{\eta}^i_q, \boldsymbol{\eta}^i_u\}_{i=1,\ldots,N_{\text{data}}}\,, \tag{A.6}$$

in which the PCA coordinates $\boldsymbol{\eta}^i_q \in \mathbb{R}^{55}$ and $\boldsymbol{\eta}^i_u \in \mathbb{R}^{21}$ are such that

$$\begin{aligned} \boldsymbol{\eta}^i_q &= [\Sigma_q]^{-1}[U_q]^T \left(\mathbf{q}^i - \overline{\mathbf{q}^1}\right)\,, \\ \boldsymbol{\eta}^i_u &= [\Sigma_u]^{-1}[U_u]^T \left(\mathbf{u}^i - \overline{\mathbf{u}^1}\right)\,, \end{aligned} \tag{A.7}$$

with $\overline{\mathbf{q}^1}$ (resp. $\overline{\mathbf{u}^1}$) the average displacement vector (resp. uncontrolled parameter) over the $N_{\text{data},1} = 3{,}342{,}948$ data points of the ANN training subset $\mathcal{D}_{\text{ANN},1}$. The ANN is thus constructed as a surrogate to the nonlinear application $(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u) \mapsto \widehat{\boldsymbol{\mathcal{F}}}^{\text{NL}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u)$ from $\mathbb{R}^{76}$ to $\mathbb{R}^{152}$, such that

$$\widehat{\boldsymbol{\mathcal{F}}}^{\text{NL}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u) = \boldsymbol{\mathcal{F}}^{\text{NL}}(\overline{\mathbf{q}^1} + [U_q][\Sigma_q]\boldsymbol{\eta}_q\,,\, \overline{\mathbf{u}^1} + [U_u][\Sigma_u]\boldsymbol{\eta}_u)\,. \tag{A.8}$$

Regarding the ANN derivative $(\mathbf{q}, \mathbf{u}) \mapsto [\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})]$ such that $[\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})]_{ij} = \frac{\partial}{\partial q_j} \mathcal{F}_i^{\text{ANN}}(\mathbf{q}, \mathbf{u})$, it can be obtained according to the chain rule, such that

$$[\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})] = [\mathrm{D}_{\boldsymbol{\eta}_q} \widehat{\boldsymbol{\mathcal{F}}}^{\text{ANN}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u)][\Sigma_q]^{-1}[U_q]^T\,, \tag{A.9}$$

where, similarly to Eq. (A.8), $\widehat{\boldsymbol{\mathcal{F}}}^{\text{ANN}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u)$ denotes the ANN surrogate of $\widehat{\boldsymbol{\mathcal{F}}}^{\text{NL}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u)$, such that

$$\widehat{\boldsymbol{\mathcal{F}}}^{\text{ANN}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u) = \boldsymbol{\mathcal{F}}^{\text{ANN}}(\overline{\mathbf{q}^1} + [U_q][\Sigma_q]\boldsymbol{\eta}_q\,,\, \overline{\mathbf{u}^1} + [U_u][\Sigma_u]\boldsymbol{\eta}_u)\,. \tag{A.10}$$

It should be noted that the Jacobian matrix $[\mathrm{D}_{\boldsymbol{\eta}_q} \widehat{\boldsymbol{\mathcal{F}}}^{\text{ANN}}(\boldsymbol{\eta}_q, \boldsymbol{\eta}_u)]$ is evaluated by neural network backpropagation, which then yields the Jacobian matrix $[\mathrm{D}_{\mathbf{q}} \boldsymbol{\mathcal{F}}^{\text{ANN}}(\mathbf{q}, \mathbf{u})]$ as given in Eq. (A.9).

Appendix B. Highlighting the nonlinear effects

The proposed ANN hyperreduction method is devoted to replacing the costly evaluation of the nonlinear forces and their derivatives with respect to displacements. The results shown in Section 6 demonstrate the very good accuracy of the proposed approach. It is thus important to make sure that in this application, the nonlinear forces are significant. To this end, the same six figures, as already presented in Section 6.6 (or as those presented in the begining of Section 6.8), are presented to highlight the difference between the nonlinear response (depicted in black) and the linear response (depicted in blue). The results are as follows. The global quadratic errors in time and frequency are $\epsilon^{\text{time}} = 135.41\%$ and $\epsilon^{\text{freq}} = 71.82\%$, while the global overlaps in time and frequency are $\text{OVL}^{\text{time}} = 27.9\%$ and $\text{OVL}^{\text{freq}} = 13.5\%$. This shows the very significant difference between the nonlinear response and the linear response, as illustrated in Figures B.22 to B.27.

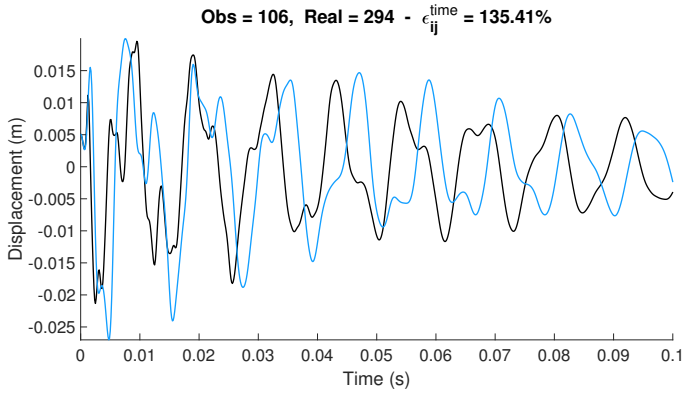

Figure B.22: Time response for observation $i = 106$ and realization $j = 294$ (among $n_{\text{train}} = 384$), for which $\epsilon_{ij}^{\text{time}} \simeq \epsilon^{\text{time}} \simeq 1.3541$. Nonlinear response (black) and linear response (blue).

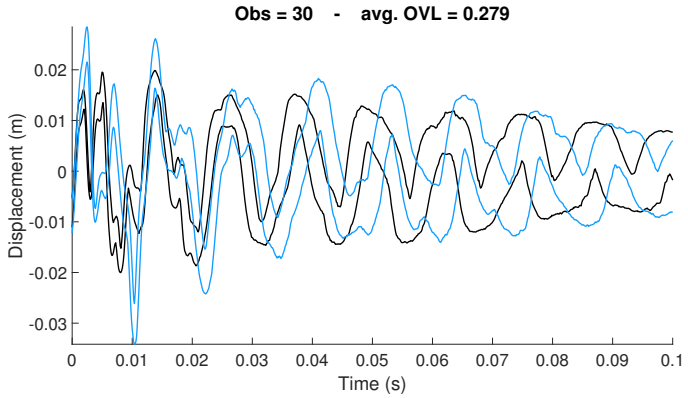

Figure B.23: Random time response (95% confidence region over $n_{\text{train}} = 384$ realizations), for observation $i = 30$, for which $\overline{\text{OVL}_i}^{\text{time}} \simeq \text{OVL}^{\text{time}} \simeq 0.279$. Nonlinear response (black) and linear response (blue).

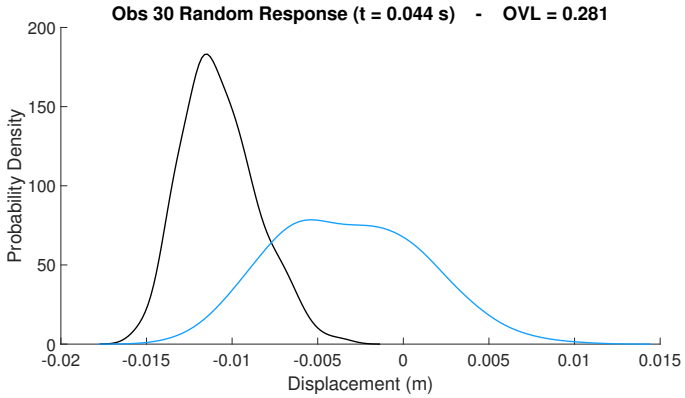

Figure B.24: Time response PDF (GKDE with $n_{\text{train}} = 384$ realizations), at instant $t_{ex} = 0.044$ s and for observation $i = 30$, for which $\text{OVL}_i^{\text{time}}(t_{ex}) \simeq \text{OVL}^{\text{time}} \simeq 0.281$. Nonlinear response (black) and linear response (blue).

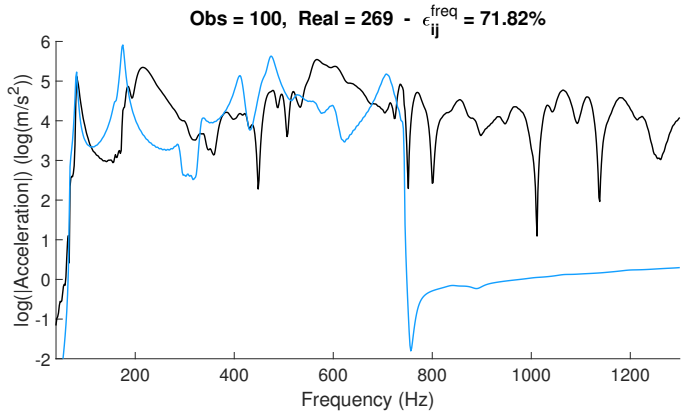

Figure B.25: Frequency response for observation $i = 100$ and realization $j = 269$ (among $n_{\text{train}} = 384$), for which $\epsilon_{ij}^{\text{freq}} \simeq \epsilon^{\text{freq}} \simeq 0.7182$. Nonlinear response (black) and linear response (blue).

Appendix C. Comparison with the nonparametric conditional estimation

Appendix C.1. Conditional response and estimation methods

The proposed ANN surrogate model enables the construction of a hyperreduced parameterized stochastic model capable of computing the random dynamical response for any given control parameter $\mathbf{w}_0 \in C_w$, with the random uncontrolled parameter $\mathbf{U}$ representing the stochastic germs responsible for the randomness. In particular, an MC simulation with random realizations of $\mathbf{U}$, using the proposed hyperreduced model with fixed $\mathbf{w}_0$, yields the conditional random response given $\mathbf{w}_0$. An alternative approach to estimate the conditional random response given $\mathbf{w}_0$ is nonparametric conditional estimation, defined as follows.

Let Q denote the random response at a given time or frequency, with values in $\mathbb{R}$. It is assumed that n realizations $\{q^\ell\}_{\ell=1,\ldots,n}$ of Q are available, with associated realizations $\{\mathbf{w}^\ell\}_{\ell=1,\ldots,n}$ of random

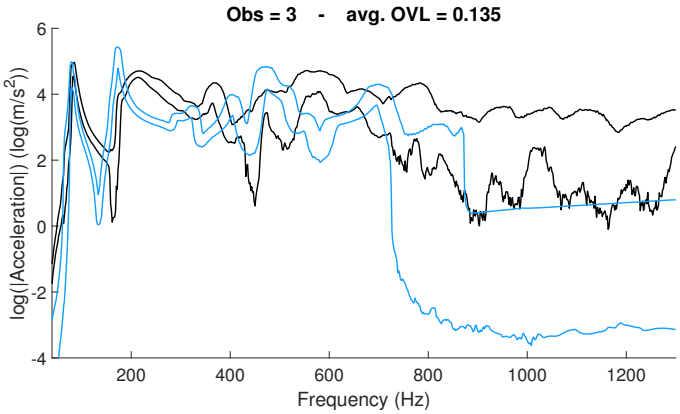

Figure B.26: Random frequency response (95% confidence region over $n_{\text{train}} = 384$ realizations), for observation $i = 3$, for which $\overline{\text{OVL}}_i^{\text{freq}} \simeq \text{OVL}^{\text{freq}} \simeq 0.135$. Nonlinear response (black) and linear response (blue).

control parameter $\mathbf{W}$. The conditional PDF of Q given $\mathbf{W}$ is denoted $p_{Q|\mathbf{W}}$ and is estimated according to GKDE of nonparametric statistics, as

$$p_{Q|\mathbf{W}}(q|\mathbf{w_0}) = \frac{1}{\sigma_q \sqrt{2\pi} s_q} \times \frac{\displaystyle\sum_{\ell=1}^{n} \exp\left(-\frac{1}{2s_q^2\sigma_q^2}(q-q^\ell)^2 - \frac{1}{2s_w^2}\sum_{k=1}^{n_w}\left(\frac{w_{0,k}-w_k^\ell}{\sigma_{w_k}}\right)^2 \right)}{\displaystyle\sum_{\ell=1}^{n} \exp\left(-\frac{1}{2s_w^2}\sum_{k=1}^{n_w}\left(\frac{w_{0,k}-w_k^\ell}{\sigma_{w_k}}\right)^2 \right)}, \tag{C.1}$$

in which σ_q and σ_{w_k} are the estimated standard deviations of Q and W_k, and where s_q and s_w are the corrected Silverman bandwidths such that:

$$s_q = \alpha_q \left(\frac{4}{n(2+n_q)}\right)^{\frac{1}{n_q+4}}, \quad s_w = \alpha_w \left(\frac{4}{n(2+n_w)}\right)^{\frac{1}{n_w+4}}, \tag{C.2}$$

with $n_q = 1$ and $n_w = 3$, and where α_q and α_w are correction factors. This Appendix C is devoted to the comparison of the following two methods for estimating the conditional random response given $\mathbf{w}_0$:

- the nonparametric estimation of the conditional probability density function of Q given $\mathbf{W} = \mathbf{w}_0$, computed with Eq. (C.1),
- the so-called direct estimation by using the proposed HROM with fixed $\mathbf{W} = \mathbf{w}_0$.

Just like in the application presented in Section 6, the admissible set C_w for the random control parameter $\mathbf{W}$ is such that

$$C_w = [0.0045, 0.0055] \times [63, 77] \times [720, 880]\,. \tag{C.3}$$

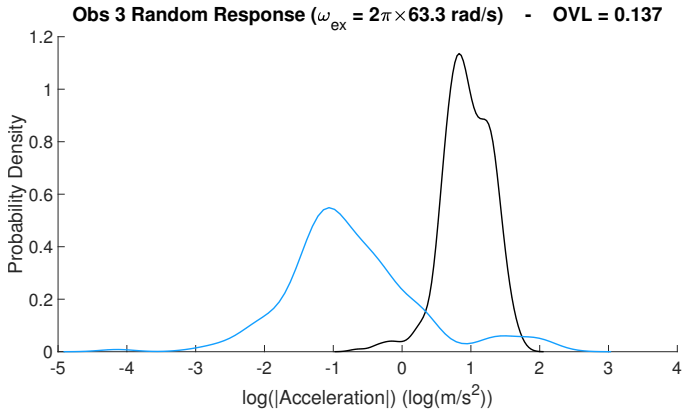

Figure B.27: Frequency response PDF (GKDE with $n_{\text{train}} = 384$ realizations), at frequency $\omega_{ex} = 2\pi \times 63.3$ rad/s and for observation $i = 3$, for which $\text{OVL}_i^{\text{freq}}(\omega_{ex}) \simeq \text{OVL}^{\text{freq}} \simeq 0.137$. Nonlinear response (black) and linear response (blue).

For the subsequent analysis of this Appendix C, the control parameter $\mathbf{w}_0 \in C_w$ is chosen as

$$\mathbf{w}_0 = (0.0053\,,\ 66\,,\ 850)\,. \tag{C.4}$$

Appendix C.2. Nonparametric estimation and direct estimation using the best hyperreduced model

For comparing the statistical convergence of the two methods mentioned in Appendix C.1, a large number of realizations is required. As demonstrated in Section 6, the HROM based on the ANN trained on $n_r = 334$ realizations exhibits both excellent accuracy and efficiency. In the sequel, it is referred to as the *best HROM*. Subsequently, such best HROM is used to generate a very large number n of MC realizations, referred to as *HROM realizations*.

Let the notation HROM($n_r = 334$) refer to such best HROM, for which the ANN was trained on $n_r = 334$ realizations. The reference estimation of the conditional random response is obtained by generating $n = 24{,}640$ HROM realizations using the best HROM with fixed $\mathbf{W} = \mathbf{w}_0$. This large number n of HROM realizations were computed in about 30 hours using the workstation with 112 CPU cores and 3 GPU nodes. The $n = 24{,}640$ HROM realizations of the best HROM with fixed $\mathbf{W} = \mathbf{w}_0$ are denoted as $\{\text{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}$. In the remainder of this appendix, we call *direct estimation* when the conditional response is performed without using Eq. (C.1) and instead, by directly fixing $\mathbf{W} = \mathbf{w}_0$. In particular, the direct estimation of the conditional random response given $\mathbf{W} = \mathbf{w}_0$, is denoted as

$$\text{Direct}\left(\{\text{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

and constitutes the reference estimation of the conditional random response given $\mathbf{W} = \mathbf{w}_0$. It is directly obtained using $n = 24,640$ MC realizations of the best HROM with fixed $\mathbf{W} = \mathbf{w}_0$.

The best HROM was also used to compute $n = 24{,}640$ HROM realizations with random $\mathbf{W}$ which, similarly, are denoted as $\{\text{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W})}$. Such computation also required

about 30 hours using the workstation with 112 CPU cores and 3 GPU nodes. Then, the nonparametric conditional estimation according to Eq. (C.1) of the random response given $\mathbf{W} = \mathbf{w}_0$ with such realizations is denoted as

$$\text{NP-Cond}\left(\{\text{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W})} \,|\, \mathbf{W} = \mathbf{w}_0\right) .$$

Let $\text{OVL}_A^{\text{time}}$ and $\text{OVL}_A^{\text{freq}}$ denote the global overlaps between the conditional random responses estimated through those two methods. We find $\text{OVL}_A^{\text{time}} \simeq 0.951$ and $\text{OVL}_A^{\text{freq}} \simeq 0.930$. The next four Figures C.28 to C.31 compare these two random responses through some of their confidence regions (for given observations) as well as through some of their PDF (for given time and frequency), which exhibit a similar overlap as the global overlaps (just like it was done in Section 6). From these figures and at this stage, it can be concluded that the nonparametric

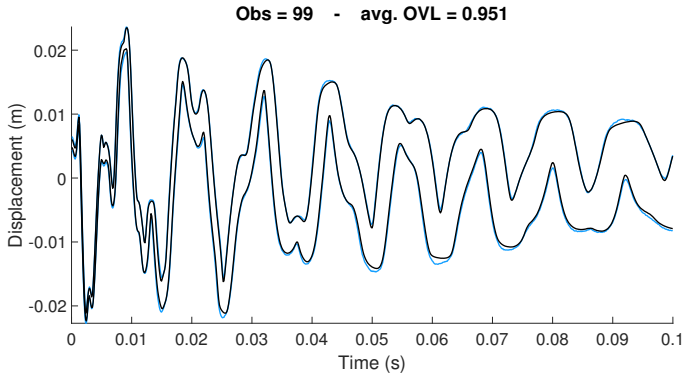

Figure C.28: Conditional random time response given $\mathbf{w}_0$ (96% confidence region over $n = 24{,}640$ HROM realizations), for observation $i = 99$, for which $\overline{\text{OVL}_i}^{\text{time}} \simeq \text{OVL}_A^{\text{time}} \simeq 0.951$. Nonparametric estimation (blue) and reference direct estimation (black).

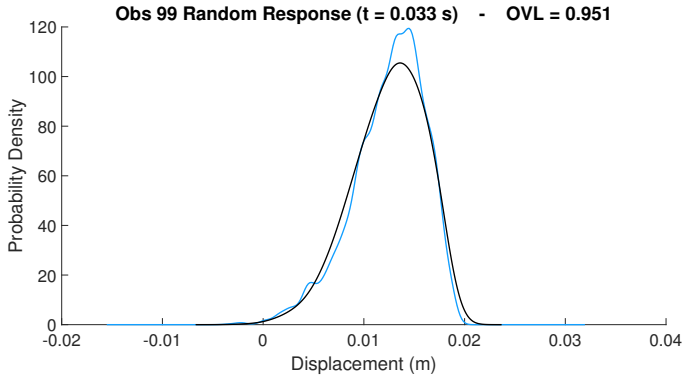

Figure C.29: Conditional time response PDF given $\mathbf{w}_0$ (GKDE with $n = 24{,}640$ HROM realizations), at time $t_{ex} = 0.033$ s and for observation $i = 99$, for which $\text{OVL}_i^{\text{time}}(t_{ex}) \simeq \text{OVL}_A^{\text{time}} \simeq 0.951$. Nonparametric estimation (blue) and reference direct estimation (black).

estimation has not yet fully converged, as indicated by the remaining discrepancies relative to the

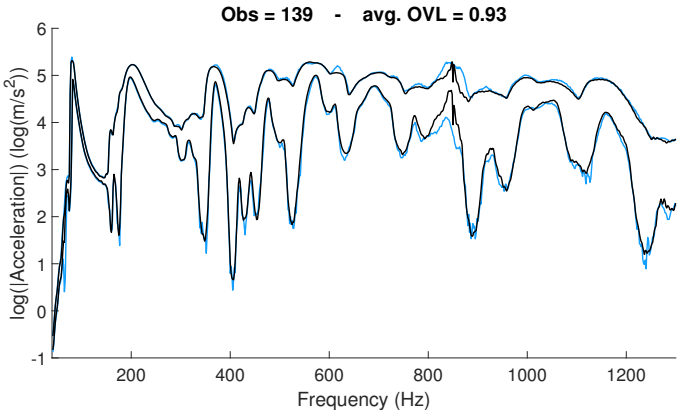

Figure C.30: Conditional random frequency response given $\mathbf{w}_0$ (96% confidence region over $n = 24{,}640$ HROM realizations), for observation $i = 139$, for which $\overline{\mathrm{OVL}}_i^{\mathrm{freq}} \simeq \mathrm{OVL}_A^{\mathrm{freq}} \simeq 0.930$. Nonparametric estimation (blue) and reference direct estimation (black).

reference direct estimation. Also, there are some oscillations in the frequency response PDF in Figure C.31, which can be explained by the choice of bandwidths through the correction factors α_q and α_w defined in Eq. (C.2). Increasing the correction factors α_q and α_w would remove such oscillations but the overlaps would deteriorate. Let α denote the correction factor affected to the Silverman bandwidth and that is used for the reference direct estimation. For the time response, the correction factors α_q^{time}, α_w^{time}, and α^{time} were optimized through a grid search so as to maximize the overlap $\overline{\mathrm{OVL}}_i^{\mathrm{time}}$ for observation $i = 122$. Likewise, for the frequency response, the correction factors α_q^{freq}, α_w^{freq}, and α^{freq} were also optimized through a grid search, so as to maximize the overlap $\overline{\mathrm{OVL}}_i^{\mathrm{freq}}$ for the same observation $i = 122$. The results are as follows,

$$\begin{aligned} &\alpha_q^{\mathrm{time}} = 1.32\,, \quad \alpha_w^{\mathrm{time}} = 1.59\,, \quad \alpha^{\mathrm{time}} = 2.1\,, \quad \overline{\mathrm{OVL}}_i^{\mathrm{time}} \simeq 0.953\,, \\ &\alpha_q^{\mathrm{freq}} = 1.34\,, \quad \alpha_w^{\mathrm{freq}} = 1.16\,, \quad \alpha^{\mathrm{freq}} = 2.86\,, \quad \overline{\mathrm{OVL}}_i^{\mathrm{freq}} \simeq 0.929\,. \end{aligned} \tag{C.5}$$

These bandwidth correction factor values were used for all subsequent GKDE estimations throughout this appendix.

Appendix C.3. Comparison between the conditional and unconditional random responses

The unconditional random response is now compared to the (reference) conditional random response. This allows us to assess the impact of fixing $\mathbf{W} = \mathbf{w}_0$, and to evaluate whether the discrepancies observed in the previous figures for the nonparametric conditional estimation remain small in comparison. The unconditional random response is denoted as

$$\mathrm{Direct}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W})}\right).$$

Let $\mathrm{OVL}_B^{\mathrm{time}}$ and $\mathrm{OVL}_B^{\mathrm{freq}}$ denote the global overlaps between the (reference) conditional random response and the unconditional random response. We find $\mathrm{OVL}_B^{\mathrm{time}} \simeq 0.747$ and $\mathrm{OVL}_B^{\mathrm{freq}} \simeq 0.690$. The next four Figures C.32 to C.35 compare these two random responses through some of their confidence regions (for given observations) as well as through some of their PDF (for given

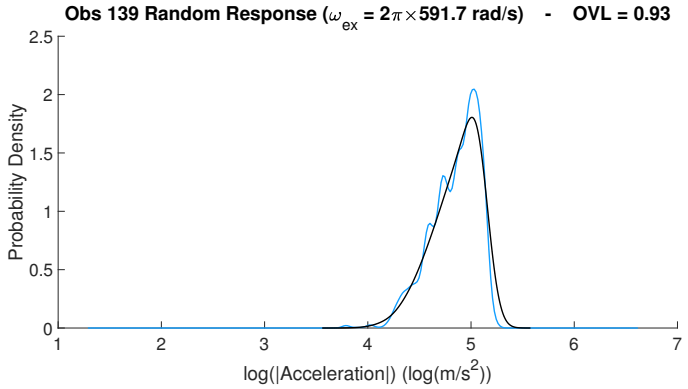

Figure C.31: Conditional frequency response PDF given $\mathbf{w}_0$ (GKDE with n = 24,640 HROM realizations), at frequency $\omega_{ex} = 2\pi \times 591.7$ rad/s and for observation $i = 139$, for which $\mathrm{OVL}_i^{\mathrm{freq}}(\omega_{ex}) \simeq \mathrm{OVL}_A^{\mathrm{freq}} \simeq 0.930$. Nonparametric estimation (blue) and reference direct estimation (black).

time and frequency), which exhibit a similar overlap as the global overlaps (in other words, the figures are similar as previously). As also attested by the overlap values, significant

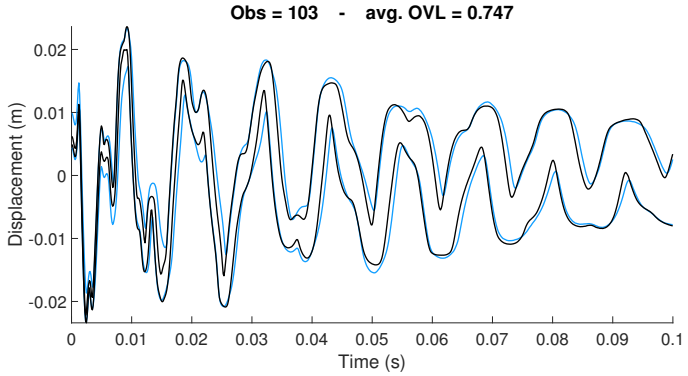

Figure C.32: Time response comparison (96% confidence region over n = 24,640 HROM realizations), for observation $i = 103$, for which $\overline{\mathrm{OVL}_i}^{\mathrm{time}} \simeq \mathrm{OVL}_B^{\mathrm{time}} \simeq 0.747$. Unconditional response (blue) and conditional response (black).

differences can be observed in Figures C.32 to C.35, which highlights the clear effect of fixing $\mathbf{W} = \mathbf{w}_0$. It can also be seen that for the most part, the conditional confidence region is included in the unconditional confidence region. Last, in the light of such comparison, the nonparametric conditional estimation with n = 24,640 HROM realizations presented in previous Appendix C.2 can be considered relatively satisfactory.

Appendix C.4. Statistical convergence analysis

Appendix C.4.1. Convergence of the direct estimation

The following two random responses are compared, namely the reference conditional response, denoted as

$$\mathrm{Direct}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

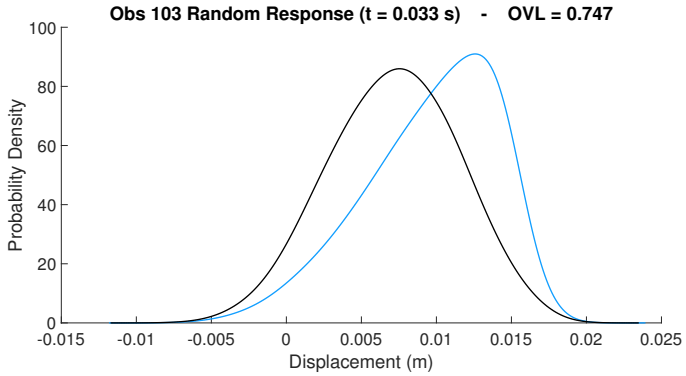

Figure C.33: Time response PDF comparison (GKDE with $n = 24{,}640$ HROM realizations), at time $t_{ex} = 0.033$ s and for observation $i = 103$, for which $\mathrm{OVL}_i^{\mathrm{time}}(t_{ex}) \simeq \mathrm{OVL}_B^{\mathrm{time}} \simeq 0.747$. Unconditional response (blue) and conditional response (black).

and the corresponding conditional response with decreased number $\nu \leq 24{,}640$ of HROM realizations used for the estimation, denoted as

$$\mathrm{Direct}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=\nu;\mathbf{W}=\mathbf{w}_0)}\right),$$

and from which the overlap $\mathrm{OVL}_C^{\mathrm{time}}(\nu)$ is estimated. The function $\nu \mapsto \mathrm{OVL}_C^{\mathrm{time}}(\nu)$ allows to describe the MC convergence of the conditional response obtained by fixing $\mathbf{W} = \mathbf{w}_0$ and with respect to the number $\nu \leq 24{,}640$ of HROM realizations.

Appendix C.4.2. Convergence of the nonparametric conditional estimation

Next, the following two random responses are compared, namely the reference conditional response, denoted as

$$\mathrm{Direct}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

and the nonparametric conditional estimation of the response as defined in Eq. (C.1) but with decreased number $\nu \leq 24{,}640$ of HROM realizations, which is denoted as

$$\text{NP-Cond}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=\nu;\mathbf{W})} \,|\, \mathbf{W} = \mathbf{w}_0\right),$$

and from which the overlap $\mathrm{OVL}_D^{\mathrm{time}}(\nu)$ is estimated. The function $\nu \mapsto \mathrm{OVL}_D^{\mathrm{time}}(\nu)$ allows to describe the convergence of the nonparametric conditional estimation with respect to the number $\nu \leq 24{,}640$ of HROM realizations.

Appendix C.4.3. Convergence of the hyperreduced model

Let $\{\mathrm{ROM}\}_{(n=n_{\mathrm{train},3}=50;\mathbf{W})}$ denote the $n = n_{\mathrm{train},3} = 50$ realizations of the ROM response, corresponding to the ANN test subset $\mathcal{D}_{\mathrm{ANN},3}$ and already presented in Section 6.6. Let then

$$\{\mathrm{HROM}(n_r = \nu)\}_{(n=n_{\mathrm{train},3}=50;\mathbf{W})}$$

denote the corresponding realizations obtained by $\{\mathrm{HROM}(n_r = \nu)\}$, namely the HROM whose ANN was trained on $n_r = \nu \leq 334$ realizations only. The following random responses are then

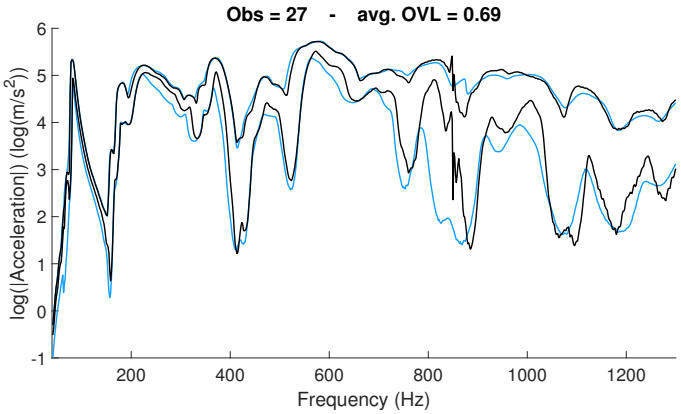

Figure C.34: Frequency response comparison (96% confidence region over $n = 24{,}640$ HROM realizations), for observation $i = 27$, for which $\overline{\text{OVL}}_i^{\text{freq}} \simeq \text{OVL}_B^{\text{freq}} \simeq 0.690$. Unconditional response (blue) and conditional response (black).

compared. Namely, the reference response calculated with the ROM, denoted as

$$\text{Direct}\left(\{\text{ROM}\}_{(n=n_{\text{train},3}=50;\mathbf{W})}\right),$$

and the approximated response calculated by the HROM with an ANN trained on $\nu \leq 334$ realizations only, denoted as

$$\text{Direct}\left(\{\text{HROM}(n_r = \nu)\}_{(n=n_{\text{train},3}=50;\mathbf{W})}\right),$$

and from which the overlap $\text{OVL}_E^{\text{time}}(\nu)$ is estimated. The function $\nu \mapsto \text{OVL}_E^{\text{time}}(\nu)$ allows to describe the convergence of the HROM with respect to the number $\nu \leq 334$ of realizations used to train its ANN. The function $\nu \mapsto \text{OVL}_E^{\text{time}}(\nu)$ has already been plotted in Figure 19 in Section 6.8.

Appendix C.4.4. Hyperreduced model with neural network trained on fifty-three realizations

The following two random responses are compared, namely the reference conditional response, denoted as

$$\text{Direct}\left(\{\text{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

and the approximated conditional response from using the HROM whose ANN was trained on $n_r = 53$ realizations only, denoted as

$$\text{Direct}\left(\{\text{HROM}(n_r = 53)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

and from which the overlaps $\text{OVL}_F^{\text{time}}$ and $\text{OVL}_F^{\text{freq}}$ are estimated. The overlap values found are $\text{OVL}_F^{\text{time}} \simeq 0.963$ and $\text{OVL}_F^{\text{freq}} \simeq 0.928$. Once again, such computation also required about 30 hours using the workstation with 112 CPU cores and 3 GPU nodes.

Appendix C.4.5. Convergence curves in time and frequency

Figure C.36 presents the following convergence curves:

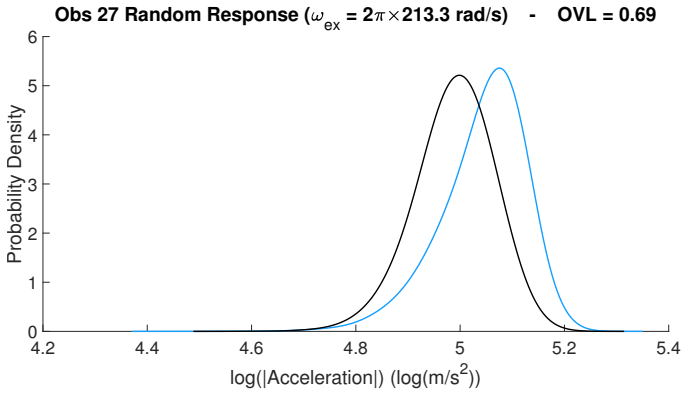

Figure C.35: Frequency response PDF comparison (GKDE with $n = 24{,}640$ HROM realizations), at frequency $\omega_{ex} = 2\pi \times 213.3$ rad/s and for observation $i = 27$, for which $\mathrm{OVL}_i^{\mathrm{freq}}(\omega_{ex}) \simeq \mathrm{OVL}_B^{\mathrm{freq}} \simeq 0.690$. Unconditional response (blue) and conditional response (black).

- $\nu \mapsto \mathrm{OVL}_C^{\mathrm{time}}(\nu)$ that pertains to the convergence of the direct estimation of the conditional response with fixed $\mathbf{W} = \mathbf{w}_0$ as defined in Appendix C.4.1 (thin line),
- $\nu \mapsto \mathrm{OVL}_D^{\mathrm{time}}(\nu)$ that pertains to the convergence of the nonparametric estimation of the conditional response with random $\mathbf{W}$ as defined in Appendix C.4.2 (dashed line),
- $\nu \mapsto \mathrm{OVL}_E^{\mathrm{time}}(\nu)$ that pertains to the convergence of the HROM response with respect to the number ν of realizations used for training its ANN and as defined in Appendix C.4.3 (thick line),

as well as the cross with coordinates $(53, \mathrm{OVL}_F^{\mathrm{time}})$ that pertains to the direct estimation of the conditional response with fixed $\mathbf{W} = \mathbf{w}_0$ using the HROM whose ANN was trained on $n_r = 53$ realizations only and as defined in Appendix C.4.4, and the horizontal line $\nu \mapsto \mathrm{OVL}_B^{\mathrm{time}}$ that pertains to the unconditional response as defined in Appendix C.3. First, it can be observed

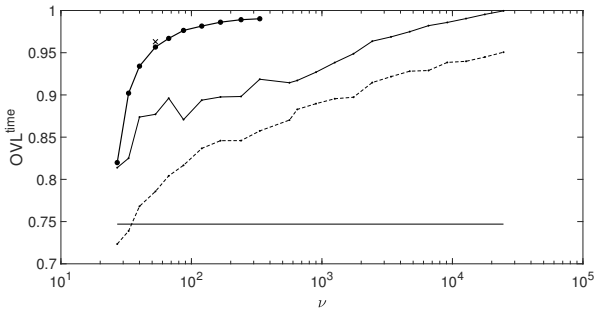

Figure C.36: Convergence curves $\nu \mapsto \mathrm{OVL}^{\mathrm{time}}(\nu)$ for the time response.

that the dashed line is far below the thin line, which demonstrates the slower convergence of the nonparametric conditional estimation as compared to the direct estimation with fixed $\mathbf{W} = \mathbf{w}_0$, which is an expected result. Next, it can be observed that the horizontal line is well below the

others, for the most part. This shows that for a sufficiently large number ν of realizations, the estimations of the conditional response given $\mathbf{w}_0$ are at least better than the unconditional response. Thus, such horizontal line serves as a baseline. Moreover, it can be seen that the cross with coordinates $(53, \mathrm{OVL}_F^{\mathrm{time}})$ is very close to the curve $\nu \mapsto \mathrm{OVL}_E^{\mathrm{time}}(\nu)$. This point $(53, \mathrm{OVL}_F^{\mathrm{time}})$ was calculated to validate the curve $\nu \mapsto \mathrm{OVL}_E^{\mathrm{time}}(\nu)$ as a relevant substitute to the convergence curve from the HROM realizations $\{\mathrm{HROM}(n_r = \nu)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}$ with $\nu \in \{27, 33, 40, 53, 67, 87, 120, 167, 240, 334\}$, which was not calculated and which would have required 30 hours for each of its ten points. The convergence curve $\nu \mapsto \mathrm{OVL}_E^{\mathrm{time}}(\nu)$ thus attests to the superiority of the HROM in providing rapid convergence of the conditional estimation with respect to the number n_r of expensive simulations required. At last, it is observed that the thin line $\nu \mapsto \mathrm{OVL}_C^{\mathrm{time}}(\nu)$ reaches unity with a significant slope, which shows that the direct estimation using $n = 24{,}640$ HROM realizations with fixed $\mathbf{W} = \mathbf{w}_0$ has not fully converged either. Nonetheless, it is still relevant as a reference for comparing the convergence of the direct estimation with fixed $\mathbf{W} = \mathbf{w}_0$ to the convergence of the nonparametric conditional estimation with random $\mathbf{W}$. Similarly to

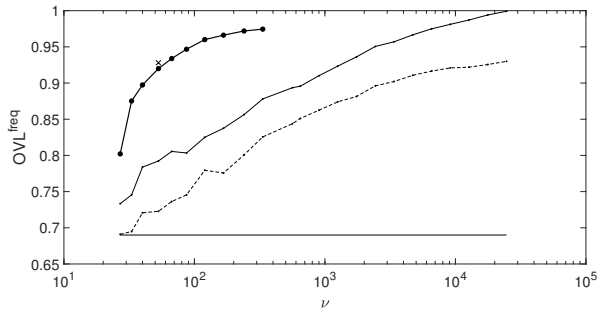

Figure C.37: Convergence curves $\nu \mapsto \mathrm{OVL}^{\mathrm{freq}}(\nu)$ for the frequency response.

Figure C.36, Figure C.37 presents the same quantities, but in frequency rather than in time. A similar behavior is observed, and the same conclusions can be drawn.

Appendix C.5. Comparison of the conditional estimations based on fifty-three expensive simulations only

Appendix C.5.1. Nonparametric estimation with random control parameter

The following two random responses are compared, namely the reference conditional response, denoted as

$$\mathrm{Direct}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=24{,}640;\mathbf{W}=\mathbf{w}_0)}\right),$$

and the nonparametric conditional estimation of the response as defined in Eq. (C.1) but with decreased number $\nu = 53 \leq 24{,}640$ of HROM realizations, which is denoted as

$$\text{NP-Cond}\left(\{\mathrm{HROM}(n_r = 334)\}_{(n=53;\mathbf{W})} \,|\, \mathbf{W} = \mathbf{w}_0\right),$$

and from which the overlaps $\mathrm{OVL}_G^{\mathrm{time}}$ and $\mathrm{OVL}_G^{\mathrm{freq}}$ are estimated. The overlap values found are $\mathrm{OVL}_G^{\mathrm{time}} \simeq 0.786$ and $\mathrm{OVL}_G^{\mathrm{freq}} \simeq 0.723$.

Based on the overlap values and Figures C.38 to C.41, it can be concluded that the nonparametric conditional estimation is not satisfactory for as few as $n_r = 53$ realizations.

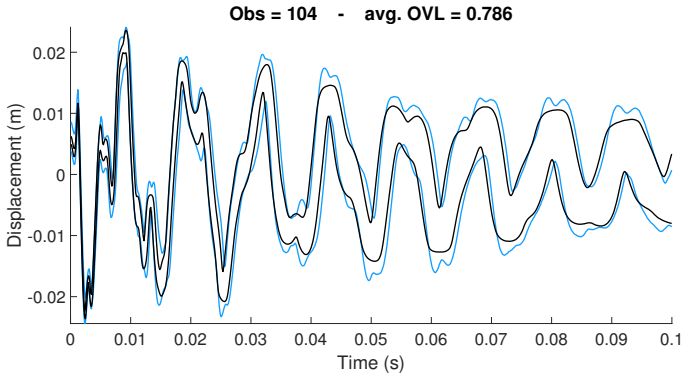

Figure C.38: Conditional random time response given $\mathbf{w}_0$ (96% confidence region), for observation $i = 104$, for which $\overline{\mathrm{OVL}}_i^{\mathrm{time}} \simeq \mathrm{OVL}_G^{\mathrm{time}} \simeq 0.786$. Nonparametric estimation with $n_r = 53$ realizations (blue) and reference direct estimation (black).

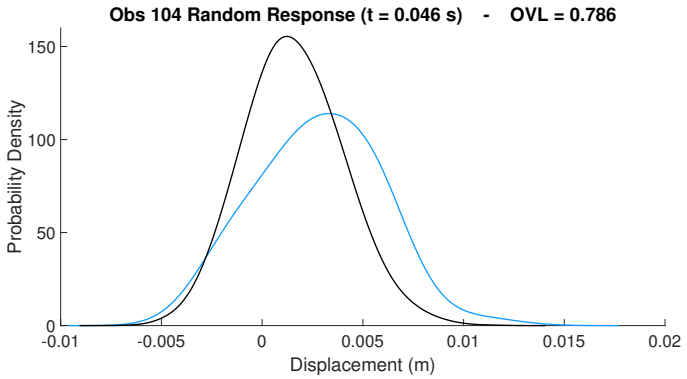

Figure C.39: Conditional time response PDF given $\mathbf{w}_0$ (GKDE), at time $t_{ex} = 0.046$ s and for observation $i = 104$, for which $\mathrm{OVL}_i^{\mathrm{time}}(t_{ex}) \simeq \mathrm{OVL}_G^{\mathrm{time}} \simeq 0.786$. Nonparametric estimation with $n_r = 53$ realizations (blue) and reference direct estimation (black).

Appendix C.5.2. Direct estimation with fixed control parameter using a hyperreduced model

As a comparison, the response from the HROM($n_r = 53$) with fixed $\mathbf{W} = \mathbf{w}_0$ is now compared to the reference conditional response. That is, the comparison as defined in Appendix C.4.4 is considered, for which the overlap values found were $\mathrm{OVL}_F^{\mathrm{time}} \simeq 0.963$ and $\mathrm{OVL}_F^{\mathrm{freq}} \simeq 0.928$. This enables a direct comparison, for the same number $n_r = 53$ of expensive simulations, between the accuracy of the conditional response estimation achieved with the proposed ANN hyperreduction method and the poor accuracy of the nonparametric conditional estimation as presented in previous Appendix C.5.1. In the Figures C.42 to C.45, the approximated conditional response that is obtained from direct estimation with fixed $\mathbf{W} = \mathbf{w}_0$ using the HROM with ANN trained on $n_r = 53$ realizations is referred to as *approximated conditional response*, for simplicity. It can be observed that the conditional response is very well estimated, which demonstrates the relevance of the machine learning strategy proposed.

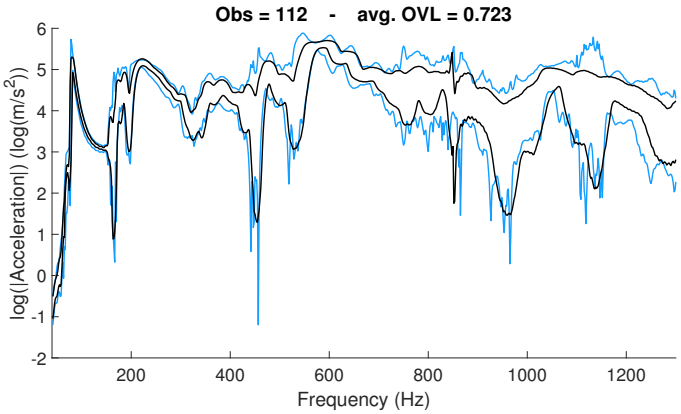

Figure C.40: Conditional random frequency response given $\mathbf{w}_0$ (96% confidence region), for observation $i = 112$, for which $\overline{\mathrm{OVL}}_i^{\mathrm{freq}} \simeq \mathrm{OVL}_G^{\mathrm{freq}} \simeq 0.723$. Nonparametric estimation with $n_r = 53$ realizations (blue) and reference direct estimation (black).

Appendix D. Criteria for updating the Jacobian matrix

In this paper, Newton–Raphson algorithm is used to solve the nonlinear equations. To enhance performance, the Jacobian matrix is not necessarily updated at each iteration or at each time step. In this appendix, the algorithm governing Jacobian updates is given. The empirical algorithm presented below is motivated by the application at hand, in which the nonlinear solver typically converges in two iterations.

Let $n_{\mathrm{iter}}^{\mathrm{exp}}$ denote the expected number of Newton iterations throughout the time simulation (for instance, $n_{\mathrm{iter}}^{\mathrm{exp}} = 2$ is a good choice for the application presented). Let i_{iter} denote the current iteration index for this time step. Let n_{add} denote the current number of iterations of the current time step that exceed the expected number $n_{\mathrm{iter}}^{\mathrm{exp}}$ of iterations, such that

$$n_{\mathrm{add}} = i_{\mathrm{iter}} - n_{\mathrm{iter}}^{\mathrm{exp}} . \tag{D.1}$$

Let n_{prev} denote the sum of numbers n_{add} from previous time steps of exceeding iterations and that occured since the last Jacobian update. Introducing $n_{\mathrm{exc}} = n_{\mathrm{add}} + n_{\mathrm{prev}}$ the total number of exceeding iterations since last Jacobian update, the first criterion for Jacobian matrix update is whenever the following condition is reached,

$$n_{\mathrm{exc}} > n_{\mathrm{exc}}^{\mathrm{max}} , \tag{D.2}$$

in which $n_{\mathrm{exc}}^{\mathrm{max}}$ is the maximum tolerated number of total iterations since last Jacobian update exceeding the expected number $n_{\mathrm{iter}}^{\mathrm{exp}}$ of iterations. The second criterion for Jacobian matrix update is whenever the following condition is reached,

$$i_{\mathrm{iter}} > n_{\mathrm{iter}}^{\mathrm{ud}} , \tag{D.3}$$

in which $n_{\mathrm{iter}}^{\mathrm{ud}}$ is the maximum number of allowed iterations before Jacobian update in the same time step. For the application presented in this paper, for all the nonlinear simulations, the num-

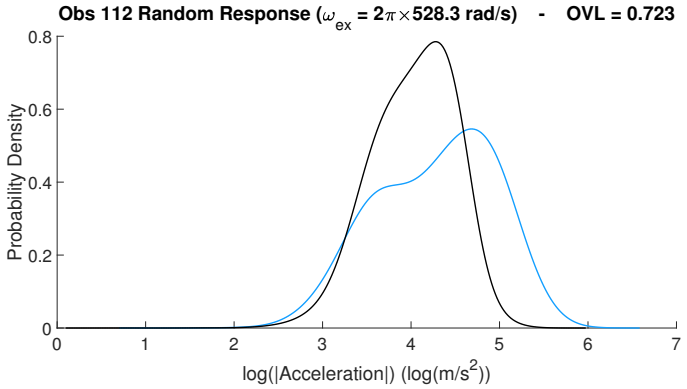

Figure C.41: Conditional frequency response PDF given $\mathbf{w}_0$ (GKDE), at frequency $\omega_{ex} = 2\pi \times 528.3$ rad/s and for observation $i = 112$, for which $\mathrm{OVL}_i^{\mathrm{freq}}(\omega_{ex}) \simeq \mathrm{OVL}_G^{\mathrm{freq}} \simeq 0.723$. Nonparametric estimation with $n_r = 53$ realizations (blue) and reference direct estimation (black).

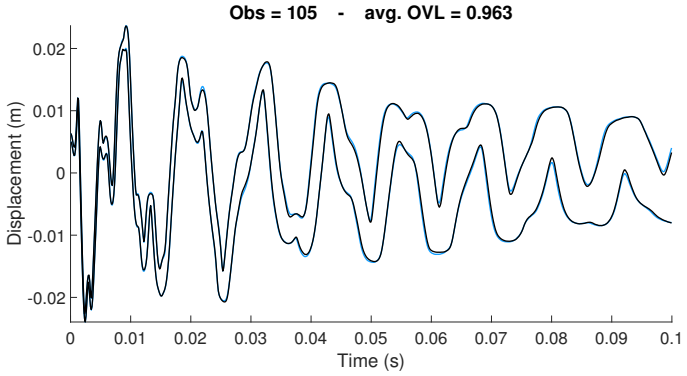

Figure C.42: Time response comparison (96% confidence region over $n = 24{,}640$ HROM realizations), for observation $i = 105$, for which $\overline{\mathrm{OVL}_i}^{\mathrm{time}} \simeq \mathrm{OVL}_F^{\mathrm{time}} \simeq 0.963$. Approximated conditional response (blue) and reference conditional response (black).

bers controlling the updating criteria were set as $n_{\mathrm{iter}}^{\mathrm{exp}} = 2$, $n_{\mathrm{exc}}^{\mathrm{max}} = 7$, and $n_{\mathrm{iter}}^{\mathrm{ud}} = 5$. Therefore, the second criterion forces a Jacobian update after five Newton iterations in the same time step, and the first criterion forces a Jacobian update whenever the number of iterations per time step grows too high as compared to the expected number of iterations. It should be noted that on average in this application, the Jacobian matrix was updated about once every forty Newton iterations.

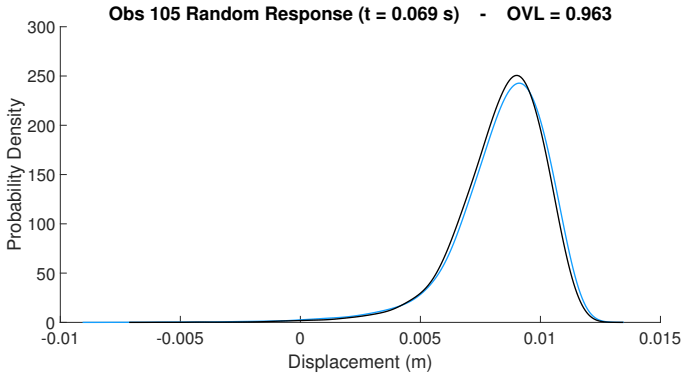

Figure C.43: Time response PDF comparison (GKDE with $n = 24{,}640$ HROM realizations), at time $t_{ex} = 0.069$ s and for observation $i = 105$, for which $\text{OVL}_i^{\text{time}}(t_{ex}) \simeq \text{OVL}_G^{\text{time}} \simeq 0.963$. Approximated conditional response (blue) and reference direct estimation (black).

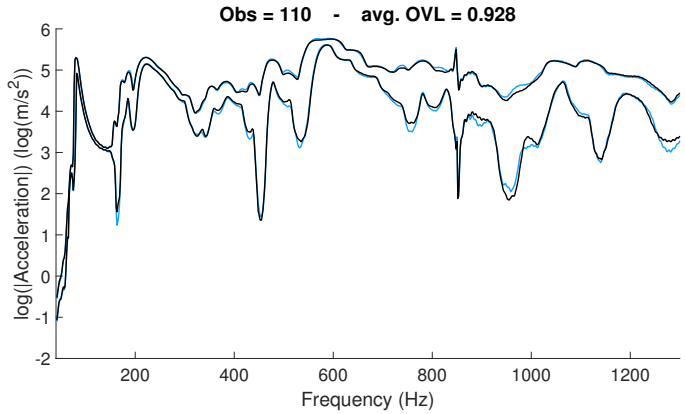

Figure C.44: Frequency response comparison (96% confidence region over $n = 24{,}640$ HROM realizations), for observation $i = 110$, for which $\overline{\text{OVL}}_i^{\text{freq}} \simeq \text{OVL}_F^{\text{freq}} \simeq 0.928$. Approximated conditional response (blue) and conditional response (black).

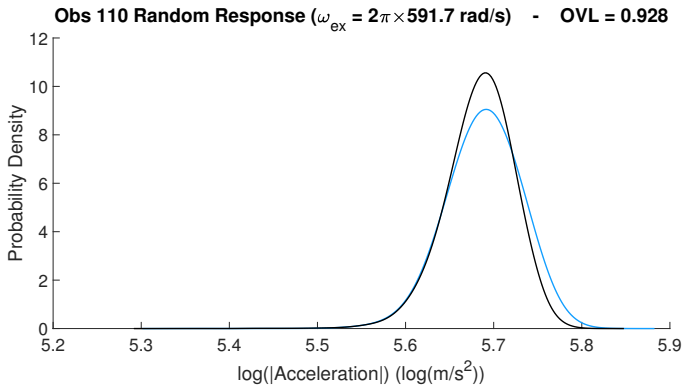

Figure C.45: Frequency response PDF comparison (GKDE with $n = 24{,}640$ HROM realizations), at frequency $\omega_{ex} = 2\pi \times 591.7$ rad/s and for observation $i = 110$, for which $\mathrm{OVL}_i^{\mathrm{freq}}(\omega_{ex}) \simeq \mathrm{OVL}_F^{\mathrm{freq}} \simeq 0.928$. Approximated conditional response (blue) and reference direct estimation (black).

References

[1] O. Huang, S. Saha, J. Guo, W. K. Liu, An introduction to kernel and operator learning methods for homogenization by self-consistent clustering analysis, Computational Mechanics 72 (1) (2023) 195–219. doi:10.1007/s00466-023-02331-w.

[2] P. Batlle, M. Darcy, B. Hosseini, H. Owhadi, Kernel methods are competitive for operator learning, Journal of Computational Physics 496 (2024) 112549. doi:10.1016/j.jcp.2023.112549.

[3] H. Eivazi, S. Wittek, A. Rausch, Nonlinear model reduction for operator learning, arXiv preprint arXiv:2403.18735 (2024). doi:10.48550/arXiv.2403.18735.

[4] X. Lu, D. G. Giovanis, J. Yvonnet, V. Papadopoulos, F. Detrez, J. Bai, A data-driven computational homogenization method based on neural networks for the nonlinear anisotropic electrical response of graphene/polymer nanocomposites, Computational Mechanics 64 (2) (2019) 307–321. doi:10.1007/s00466-018-1643-0.

[5] K. Cheng, R. Zimmermann, Adaptive data-driven probabilistic reduced-order models for parameterized dynamical systems, SIAM Journal on Scientific Computing 47 (2) (2025) C426–C450. doi:10.1137/23M1602280.

[6] E. Whalen, C. Mueller, Toward reusable surrogate models: Graph-based transfer learning on trusses, Journal of Mechanical Design 144 (2) (2022) 021704. doi:10.1115/1.4052298.

[7] J. Kneifl, J. Fehr, S. L. Brunton, J. N. Kutz, Multi-hierarchical surrogate learning for explicit structural dynamical systems using graph convolutional neural networks, Computational Mechanics (2024) 1–21doi:10.1007/s00466-024-02553-6.

[8] V. Krokos, V. Bui Xuan, S. P. Bordas, P. Young, P. Kerfriden, A bayesian multiscale CNN framework to predict local stress fields in structures with microscale features, Computational Mechanics 69 (3) (2022) 733–766. doi:10.1007/s00466-021-02112-3.

[9] D. Geng, J. Yan, Q. Xu, Q. Zhang, M. Zhou, Z. Fan, H. Li, Real-time structure topology optimization using CNN driven moving morphable component method, Engineering Structures 290 (2023) 116376. doi:10.1016/j.engstruct.2023.116376.

[10] T. Ryu, J. P. Heuss, P. Haley, C. Mirabito, E. Coelho, P. Hursky, M. C. Schönau, K. Heaney, P. F. Lermusiaux, Adaptive stochastic reduced order modeling for autonomous ocean platforms, IEEE, 2021. doi:10.23919/OCEANS44145.2021.9705790.

[11] D. Pivovarov, Novel adaptive reduced order spectral non-deterministic FEM approach to problems with general interval-fuzzy-stochastic uncertainties, Computational Mechanics (2025) 1–26doi:10.1007/s00466-025-02658-6.

[12] A. P. Trischler, G. M. D'Eleuterio, Synthesis of recurrent neural networks for dynamical system simulation, Neural Networks 80 (2016) 67–78. doi:10.1016/j.neunet.2016.04.001.

[13] K. Taneja, X. He, Q. He, J.-S. Chen, A multi-resolution physics-informed recurrent neural network: formulation and application to musculoskeletal systems, Computational Mechanics 73 (5) (2024) 1125–1145. doi:10.1007/s00466-023-02403-x.

[14] C. Soize, Design optimization under uncertainties of a mesoscale implant in biological tissues using a probabilistic learning algorithm, Computational Mechanics 62 (3) (2018) 477–497. doi:10.1007/s00466-017-1509-x.

[15] X. Liu, S. Tian, F. Tao, H. Du, W. Yu, Machine learning-assisted modeling of composite materials and structures: a review, in: AIAA Scitech 2021 Forum, 2021, p. 2023. doi:10.2514/6.2021-2023.

[16] Z. Wang, A. Bhaduri, H. Xu, L. Wang, An uncertainty-aware deep learning framework-based robust design optimization of metamaterial units, Structural and Multidisciplinary Optimization 68 (3) (2025) 1–26. doi:10.1007/s00158-025-03979-z.

[17] M. A. Grepl, Y. Maday, N. C. Nguyen, A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM: Mathematical Modelling and Numerical Analysis 41 (3) (2007) 575–605. doi:10.1051/m2an:2007031.

[18] N. Nguyen, J. Peraire, An efficient reduced-order modeling approach for non-linear parametrized partial differential equations, International Journal for Numerical Methods in Engineering 76 (1) (2008). doi:10.1002/nme.2309.

[19] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (5) (2010) 2737–2764.

[20] J. Degroote, J. Virendeels, K. Willcox, Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis, International Journal for Numerical Methods in Fluids 63 (2010) 207–230. doi:10.1002/fld.2089.

[21] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, International Journal for Numerical Methods in Engineering 86 (2) (2011) 155–181. doi:10.1002/nme.3050.

[22] K. Carlberg, C. Farhat, A low-cost, goal-oriented compact proper orthogonal decomposition basis for model reduction of static systems, International Journal for Numerical Methods in Engineering 86 (3) (2011) 381–402. doi:10.1002/nme.3074.

[23] D. Amsallem, M. J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, International Journal for Numerical Methods in Engineering 92 (10) (2012) 891–916. doi:10.1002/nme.4371.
[24] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, Journal of Computational Physics 242 (2013) 623–647. doi:10.1016/j.jcp.2013.02.028.
[25] M. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1077–1110. doi:10.1002/nme.4770.
[26] D. Amsallem, M. Zahr, Y. Choi, C. Farhat, Design optimization using hyper-reduced-order models, Structural and Multidisciplinary Optimization 51 (4) (2015) 919–940. doi:10.1007/s00158-014-1183-y.
[27] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Review 57 (4) (2015) 483–531. doi:10.1137/130932715.
[28] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, Vol. 92 of UNITEXT, Springer, 2015. doi:10.1007/978-3-319-15431-2.
[29] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, Journal of Computational Physics 202 (1) (2005) 346–366. doi:10.1016/j.jcp.2004.07.015.
[30] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, International Journal for Numerical Methods in Engineering 98 (9) (2014) 625–662. doi:10.1002/nme.4668.
[31] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1077–1110. doi:10.1002/nme.4820.
[32] E. Jewell, C. Farhat, C. Soize, A tractable nonparametric probabilistic approach for modeling and quantifying model-form uncertainty in turbulent CFD, Journal of Computational Physics (2025) 114067doi:10.1016/j.jcp.2025.114067.
[33] O. M. Agudelo, J. J. Espinosa, B. De Moor, Acceleration of nonlinear POD models: A neural network approach, in: 2009 European Control Conference (ECC), 2009, pp. 1547–1552. doi:10.23919/ECC.2009.7074626.
[34] H. Gao, J.-X. Wang, M. J. Zahr, Non-intrusive model reduction of large-scale, nonlinear dynamical systems using deep learning, Physica D: Nonlinear Phenomena 412 (2020). doi:https://doi.org/10.1016/j.physd.2020.132614.
[35] L. Cicci, S. Fresca, A. Manzoni, Deep-HyROMnet: A deep learning-based operator approximation for hyper-reduction of nonlinear parametrized PDEs, Journal of Scientific Computing 93 (2) (2022). doi:10.1007/s10915-022-02001-8.
[36] D. Fleres, D. De Gregoriis, O. Atak, F. Naets, Non-intrusive parametric hyper-reduction for nonlinear structural finite element formulations, Computer Methods in Applied Mechanics and Engineering 434 (2025). doi:https://doi.org/10.1016/j.cma.2024.117532.
[37] E. Capiez-Lernout, C. Soize, M. Mbaye, Mistuning analysis and uncertainty quantification of an industrial bladed disk with geometrical nonlinearity, Journal of Sound and Vibration 356 (10) (2015) 124–143. doi:{10.1016/j.jsv.2015.07.006}.
[38] M. Crisfield, Non-linear finite element analysis of solids and structures, Vol.1: essentials, John Wiley and Sons, Chichester, 1997.
[39] R. De Borst, M. Crisfield, J. Remmers, C. V. Verhoosel, Non-Linear Finite Element Analysis of Solids and Structures, Second Edition, Wiley, 2012. doi:10.1002/9781118375938.
[40] E. Capiez-Lernout, O. Ezvan, C. Soize, Updating nonlinear stochastic dynamics of an uncertain nozzle model using probabilistic learning with partial observability and incomplete dataset, Journal of Computing and Information Science in Engineering 24 (6) (2024) 061006. doi:10.1115/1.4065312.
[41] C. Soize, Non-gaussian positive-definite matrix-valued random fields for elliptic stochastic partial differential operators, Computer Methods in Applied Mechanics and Engineering 195 (1) (2006) 26–64. doi:https://doi.org/10.1016/j.cma.2004.12.014.
[42] C. Soize, Uncertainty Quantification. An Accelerated Course with Advanced Applications in Computational Engineering, Springer, New York, 2017. doi:10.1007/978-3-319-54339-0.