MOX-Report No. 20/2024

Structural health monitoring of civil structures: A diagnostic framework powered by deep metric learning

Torzoni, M.; Manzoni, A.; Mariani, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

Structural health monitoring of civil structures: A diagnostic framework powered by deep metric learning

Matteo Torzoni[a], Andrea Manzoni[b], Stefano Mariani[a]

[a] *Dipartimento di Ingegneria Civile e Ambientale, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy*
[b] *MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy*

Abstract

Recent advances in learning systems and sensor technology have enabled powerful strategies for autonomous data-driven damage detection in structural systems. This work proposes a novel method for the real-time localization of damage relying on a Siamese convolutional neural network. The method exploits a learnable mapping of raw vibration measurements onto a low-dimensional space, wherein damage locations can be easily identified. The mapping is learned in a supervised pairwise fashion exploiting labelled data, to induce a task-specific metric that allows to encode the damage position in the structure. Training data are generated through a reduced-order numerical model of the monitored structure. The damage position is then identified by performing a regression in the resulting low-dimensional features space. The proposed method does not require to define a-priori target classes and decision boundaries, thus requiring a limited amount of user-dependent assumptions. Results relevant to an L-shaped cantilever beam and a portal frame railway bridge demonstrate that the procedure can be effectively exploited for the purpose of damage localization. The method also proves to be insensitive to operational variability, measurement noise and modeling inaccuracies.

1. Introduction

Civil structures are a backbone of our society. However, they may progressively deteriorate and accumulate damage during their service life. A proper health monitoring strategy for civil structures would greatly reduce the maintenance costs over time and also avoid potential tragic events. As the installation of permanent real-time data collecting systems has become affordable [1], systematic diagnostic and prognostic activities can nowadays allow for a condition-based maintenance approach. This evolution involves benefits related not only to users safety; indeed, traffic delays and lost productivity due to maintenance, repair, and rehabilitation of infrastructures hamper the economic growth and cost to the users dozens of billions of dollars a year only in the US [2]. For instance, a potential shutdown of the Hudson River rail tunnel in New York City has been estimated to cost the US economy about $16 billion [3], with more than half due to the time lost by users for longer daily commutes.

Email addresses: matteo.torzoni@polimi.it (Matteo Torzoni), andrea1.manzoni@polimi.it (Andrea Manzoni), stefano.mariani@polimi.it (Stefano Mariani)

Modern structural health monitoring (SHM) strategies aim at monitoring a structural system from collected sensor data, enabling informed evaluations of the current state of that system [4]. Specifically, vibration-based monitoring techniques assume the dynamic characteristics of the structure to be sensitive to damage [5], independently of its location and for a severity beyond a critical threshold to allow distinguishing its effect from environmental and operational variability. Because of their ability of easily managing big data affected by noise, vibration-based data-driven approaches [6–8] represent today the most promising solution to the SHM challenges. These procedures can take advantage of an offline/online decomposition. During the offline phase, a dataset involving conditions of interest for the monitored structure is collected and used to calibrate the algorithm, in order to get rid of all the potential issues linked to the aforementioned variability. During the online phase, the trained algorithm is exploited to produce informed estimations of the structural health. According to the statistical pattern recognition paradigm [6, 9], damage-sensitive features are extracted from the raw vibration recordings, represented for instance by acceleration or displacement multivariate time series, and processed by machine learning (ML) algorithms. The use of deep learning (DL) [10, 11] can further empower such data-driven approach. By catching temporal correlations within and across time recordings, the pattern recognition capability of DL models enables to automatically learn damage-sensitive features from the raw sensor data, without the need of a preliminary feature engineering stage. Moreover, since the underlying mapping is learned offline, the structural state can be next assessed in real time, making the DL-empowered strategies suitable for SHM stream applications.

This work specifically focuses on damage localization, with damage assumed to be already detected but not localized yet. Focusing on data-driven strategies, this problem can be addressed either in an unsupervised or a supervised way. The unsupervised case is usually carried out as an extended outlier analysis, repeated for all the outputs of the sensor system. Unsupervised solutions are sometimes preferred to supervised ones, as they only require data relevant to the damage-free condition of the structure to detect the inception of a drift from the baseline, characterizing the presence of damage. Relevant algorithms involve, for instance, one-class support vector machine [12, 13] and combinations of density-based and distance-based techniques [14, 15]. On the other hand, supervised algorithms leverage labeled data related to known damage positions in order to locate the damage itself. However, it is practically impossible to conduct destructive experiments on real civil structures, in order to obtain data involving the various damage states. Labeled data pertaining to specific damage conditions can be instead generated numerically; experimental measurements are then replaced by the responses of a physics-based digital twin of the structure to be monitored, whose damage patterns can be systematically reproduced. Such a database can then be used to train the ML model. Such an hybridization between data and physics is referred to as simulation-based paradigm [16], and can be thought of as the way to let ML models learn in a simulated environment, before transferring their knowledge to the real world [7]. For instance, recent contributions in this field have been obtained by means of fully convolutional classifiers [17–19] and decision trees [20, 21]. Parametric model order reduction strategies that enjoy an offline/online decoupling are another example of how the information from physics-based models can be exploited. As proposed in [22, 23], such a methodology can be adopted to efficiently solve vibration-based crack detection problems by feeding an outer-loop

optimization algorithm.

Convolutional neural networks (NNs) have recently been used to develop state-of-the-art learning machines for vibration-based SHM of civil structures [24–27]. Originally developed within the computer vision community, convolutional NNs have quickly become a first choice to solve several problems, outperforming alternative methods. Their success is due to: the synthesis of the feature selection and feature extraction stages into a single learning stage; the sparse connectivity and parameter sharing, which lead to a great computational efficiency in comparison to feed-forward fully-connected NNs and recurrent models, such as the long-short term memory and the gated recurrent unit; the tolerance for translation, scaling, skewing, and distortion of the input data [28]. The popularity for SHM purposes is also due to their ability to detect correlation in time, within and across time series, through repeated convolution operations [17].

A supervised method is here proposed to localize damage without the need of defining a bunch of target classes in advance – a task that often seems unreasonable. For instance, a large number of locations can be foreseen for a crack in a concrete structure, which cannot be fully accounted for within a standard classification setting. To this aim, a combination of some aspects from metric learning and dimensionality reduction is exploited in this paper. Metric learning [29, 30] aims at learning a distance function that conforms to a task-specific definition of similarity, such that the true neighbors of a testing example will be closer than non-neighbors in the learned metric space [31]. Since it is often impossible to infer the quantity of interest directly from sensors data, due to the curse of dimensionality [32], dimensionality reduction is here intended as a strategy to map the high dimensional inputs onto a low-dimensional manifold, embedding the relevant semantic content [33]. The term semantic stems from the fact that the components of the low-dimensional space encode the position of the damage in the structure (semantic content). In concrete terms, a nonlinear projection of the input data onto the feature space is devised by means of a deep NN, under the constraint of inducing a damage position-related metric in the low-dimensional space. The training stage is performed in a supervised pairwise fashion, exploiting a Siamese convolutional architecture [34], and consists in learning a smooth and ordered mapping of the labelled data. After learning the low-dimensional mapping, informed predictions of the damage position are carried out in this low-dimensional space by performing a simple regression.

To efficiently assemble a synthetic dataset, sufficiently representative of potential damage scenarios and operating conditions the structure might undergo during its lifetime, a reduced-order modeling strategy for parametrized systems is exploited. Specifically, the reduced basis (RB) method [35] is adopted to replace the full-order numerical model of the monitored structure with a cheaper, yet accurate, reduced-order model (ROM).

The strength of the proposed procedure relies in its ability of linking the distances in the projection space and those in the physical space, avoiding the definition of target classes. Moreover, the damage assessment is done without tuning highly sensitive hyperparameters, such as distance thresholds and margins, and yields easily interpretable results.

The proposed methodology is investigated by adopting three alternative loss functions to train the Siamese network, and by also varying the size of the low-dimensional feature space. Results relevant to an L-shaped cantilever beam and a portal frame railway bridge are shown

to provide notable damage localization capabilities, outperforming alternative real-time methods. Even if real experimental data are not exploited in this paper, the robustness of the method against measurement noise is assessed considering noise-contaminated measurements.

The remainder of the paper is organized as follows. The proposed methodology is detailed in Sec. 2. The application of the methodology to a couple of case studies is discussed in Sec. 3 and in Sec. 4, respectively dealing with an L-shaped cantilever beam and a portal frame railway bridge. Conclusions and future developments are finally reported in Sec. 5.

2. Method

In the proposed framework, the damage localization problem is traced back to the construction of an ordered and smooth mapping of the labelled data onto a low-dimensional space. The adopted notation and the composition of the dataset are specified in Sec. 2.1, while the NN performing the mapping alongside its training and prediction phases are described in Sec. 2.2; the numerical model and the order reduction strategy adopted to populate the synthetic dataset are finally discussed in Sec. 2.3.

2.1. Data specification

The dataset $\mathbf{D}$ is built upon I instances provided by a physics-based numerical model of the structure to be monitored, assembled as

$$\mathbf{D} = \{(\mathbf{U}_i, \boldsymbol{\mathit{y}}_i)\}_{i=1}^{I} \ . \tag{1}$$

Each instance consists of N_u raw vibration recordings, $\mathbf{u}_n \in \mathbb{R}^L$, $n = 1, \ldots, N_u$, each including L measurements, and shaped as a multivariate time series $\mathbf{U}_i(\boldsymbol{\eta}_i, \boldsymbol{\mathit{y}}_i, \delta_i) = [\mathbf{u}_1, \ldots, \mathbf{u}_{N_u}]_i \in \mathbb{R}^{N_u \times L}$, where: $\boldsymbol{\eta}_i$ is the vector of parameters ruling the operational conditions of the i-th instance; $\boldsymbol{\mathit{y}}_i$ and δ_i characterize the relevant structural damage as specified in the following. The parameters defining the operational and, possibly, the environmental conditions [36, 37] undergone by the structure while the i-th instance is collected, are condensed in the vector $\boldsymbol{\eta}_i \in \mathbb{R}^{\mathtt{N_{par}}}$, $\mathtt{N_{par}}$ being the number of parameters. Each instance thus refers to a time window $(0, T)$, $T = (L-1)/\mathscr{f}$ with $\mathscr{f}$ being the sensor sampling rate, short enough to assume steady operational, environmental, and damage conditions, yet long enough to assure capturing the mechanical response of the structure. The associated label $\boldsymbol{\mathit{y}}_i \in \mathbb{R}^D$ identifies the position of the damaged zone, so that D represents the dimension of the problem. Damage is modeled by reducing the material stiffness by a magnitude δ_i within a domain $\Omega_{\boldsymbol{\mathit{y}}}$ identified by the spatial coordinates $\boldsymbol{\mathit{y}}_i$ of its center of mass. Even though a distinct δ_i is assigned for each instance to account for uncertainties related to its value, it is not considered part of the label as only the damage localization task is of interest. In order to mimic the measurement noise, each vibration recording is corrupted by adding an independent, identically distributed Gaussian noise, whose statistical properties depend on the target accuracy of the sensors.

Situations characterized by a small damage evolution rate are considered. In this context, a stiffness reduction fixed in time stands as the simplest possible damage model resulting from a time scale separation between damage evolution and damage assessment. This is a standard assumption in the literature [6] and does not limit the type of damage patterns that can be represented, like e.g.: failure of bolted connections in steel structures [38]; crack

formation in reinforced concrete beams [39]; cracks and delaminations in aircraft wings [40]. The amplitude of stiffness reduction δ_i can be selected to populate the dataset $\mathbf{D}$ of instances representative of structural states sufficiently deteriorated with respect to the damage-free baseline, so that their effects on the measurable response can be distinguished from the noise and the changing operational conditions, yet with enough structural resources not to be fully compromised. Indeed, if damage can be detected early before reaching such critical conditions, it is possible to reduce the lifecycle costs and to increase the safety and availability of the monitored structure. To simplify the notation, the index i will be dropped in the following, if unnecessary to refer to a specific instance.

To build the synthetic dataset $\mathbf{D}$, the parametric input space of the numerical model is assumed to display a uniform probability distribution for each parameter. To sample the input parameters from the aforementioned space, the latin hypercube rule is here considered, as it ensures a good compromise between randomness and coverage of the parameters domain [41]. Probability distributions other than the uniform one can also be allowed for and will be envisaged in future developments, to account for any a-priori knowledge of the structure and of the relevant operational conditions.

2.2. Damage localization model

For any given distance measure, a feature extraction algorithm can be considered as a way to learn a similarity metric among the input data [42]. In this context, a NN is designed to approximate a target function $G : \mathbf{U} \in \mathbb{R}^{N_u \times L} \rightarrow \mathbf{h} \in \mathbb{R}^{D_h}$, mapping the vibration recordings $\mathbf{U}$ onto its representation $\mathbf{h}$ in a low-dimensional manifold of size D_h. The distance $\mathcal{E}_h = \mathcal{E}_h(\mathbf{h}_1, \mathbf{h}_2)$, computed according to a chosen metric, e.g. the cosine or the Euclidean one, between any pair of mappings $\mathbf{h}_1 = G(\mathbf{U}_1)$ and $\mathbf{h}_2 = G(\mathbf{U}_2)$, is required to approximate, at least semantically, the Euclidean distance $\mathcal{E}_{\boldsymbol{y}} = \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2$ between the associated target labels $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. A possible approach to achieve this goal relies on pairwise learning, which implies that instances are processed in pairs. By fixing a threshold distance $\overline{\mathcal{E}_{\boldsymbol{y}}}$, below which the associated damage conditions can be considered similar, each instance is paired with τ_N dissimilar instances (or negative pairs), characterized by $\mathcal{E}_{\boldsymbol{y}} > \overline{\mathcal{E}_{\boldsymbol{y}}}$, and τ_P similar instances (or positive pairs), characterized by $\mathcal{E}_{\boldsymbol{y}} \leq \overline{\mathcal{E}_{\boldsymbol{y}}}$. In this way, the dataset is augmented, and becomes

$$\mathbf{D}_P = \{(\mathbf{U}_1, \boldsymbol{y}_1, \mathbf{U}_2, \boldsymbol{y}_2)_j\}_{j=1}^{I_P} \ , \tag{2}$$

$I_P = I(\tau_P + \tau_N)$ being now the total number of pairs.

The Siamese architecture, originally proposed for signature verification [34], allows for pairwise learning and can be thought of as a network made of two heads linked by the loss function, see Fig. 1. Here, the two sub-networks share the same set of parameters $\boldsymbol{\Omega}$, and each of them is therefore simply referred to as the sub-network providing the mapping $G(\cdot)$. The vibration recordings $\mathbf{U}_1$ and $\mathbf{U}_2$ are then processed, yielding the two outputs $\mathbf{h}_1$ and $\mathbf{h}_2$; the loss function then computes the associated distance $\mathcal{E}_h$ to rule the training. In the

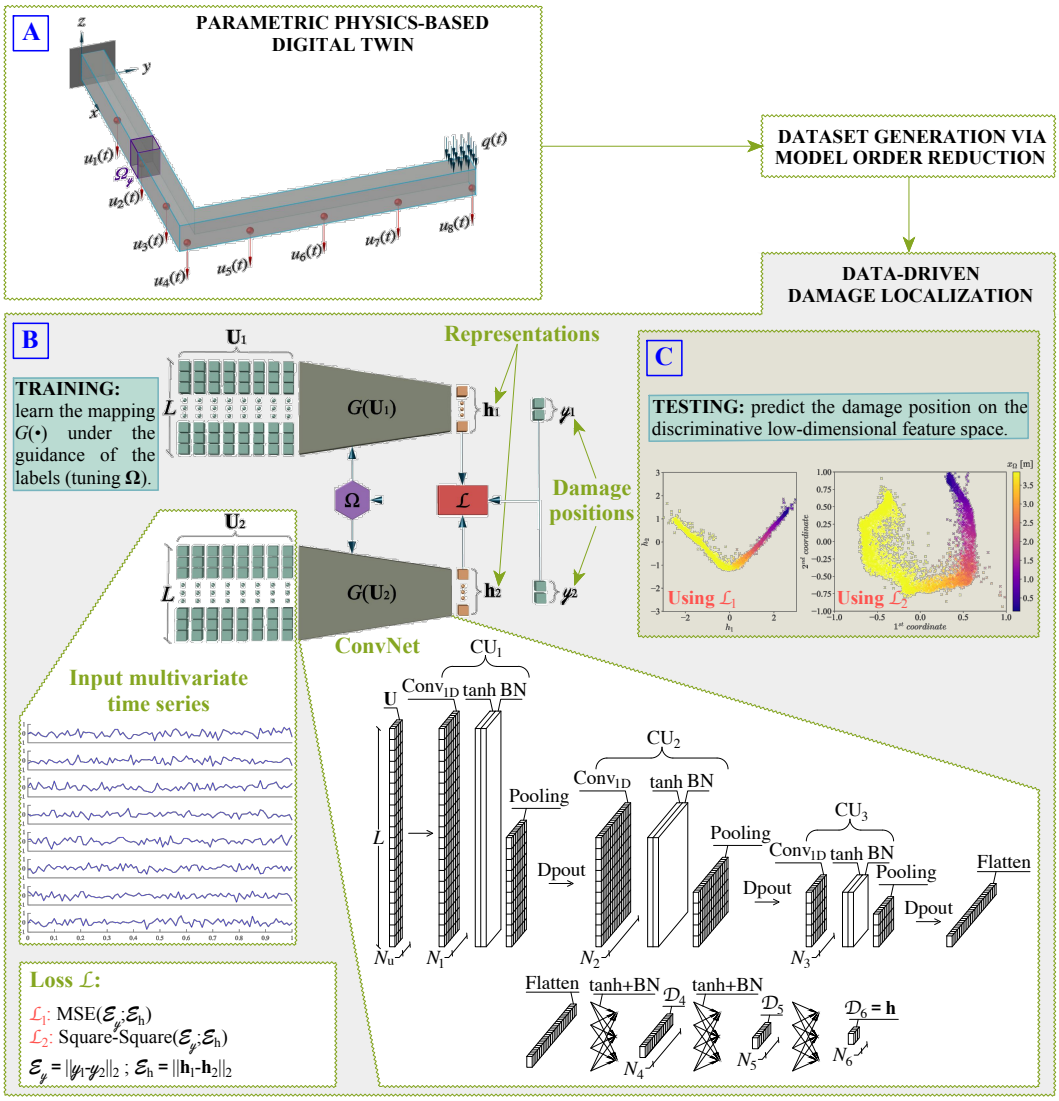


Figure 1: Scheme of the proposed methodology - A: physics-based digital twin of the monitored structure used to simulate vibration recordings in the presence of damage; model order reduction strategy to speed up the dataset population. B: NN parameters $\boldsymbol{\Omega}$, controlling mapping $G(\cdot)$, are learned by processing paired vibration recordings $\mathbf{U}_1, \mathbf{U}_2$, and minimizing a functional $\mathcal{L}$ of the distance $\mathcal{E}_h$ between the two low-dimensional representations $\mathbf{h}_1, \mathbf{h}_2$ and of the Euclidean distance $\mathcal{E}_y$ between the associated labels $\boldsymbol{y}_1, \boldsymbol{y}_2$; subnetwork $G(\cdot)$ features three 1D convolutional units CU_k, $k = 1, 2, 3$, each one provided with a 1D convolutional layer ($\mathrm{Conv_{1D}}$) ruled by tanh activation function, batch normalization (BN), 1D max pooling (Pooling) and dropout regularization (Dpout); the output is reshaped through a flatten layer and run through a stack of three fully-connected layers $\mathcal{D}_k$, $k = 4, 5, 6$, whose output is the low-dimensional representation $\mathbf{h}$; the first two fully-connected layers are activated by a tanh function before undergo a BN, while the last one has no activation function; N_k refers to the number of kernel filters for the convolutional layers and to the number of neurons for the dense layers. C: the damage position associated to vibration recordings unseen during training is predicted performing a simple regression in the low-dimensional space.

following, two alternative loss functions are considered:

$$\mathcal{L}_1(\boldsymbol{\Omega}, \mathbf{D}_P) = \frac{1}{I_P}\sum_{j=1}^{I_P}(\mathcal{E}_y - \mathcal{E}_h)_j^2 + \lambda\|\boldsymbol{\Omega}\|_2^2 \; ; \tag{3}$$

$$\mathcal{L}_2(\boldsymbol{\Omega}, \mathbf{D}_P) = \frac{1}{I_P}\sum_{j=1}^{I_P}\left\{(1-\gamma)\frac{1}{2}(\mathcal{E}_h)^2 + \gamma\frac{1}{2}\left[\max\left(0, \psi - \mathcal{E}_h\right)\right]^2\right\}_j + \lambda\|\boldsymbol{\Omega}\|_2^2 \; . \tag{4}$$

$\mathcal{L}_1$ is the mean squared error between $\mathcal{E}_h$ and $\mathcal{E}_{\boldsymbol{y}}$, while $\mathcal{L}_2$ is the pairwise contrastive loss [33, 43], or square-square loss [44]. Herein: λ is the L^2 regularization rate; $\gamma = \{0, 1\}$, respectively if $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ identify a positive pair or a negative pair; $\psi > 0$ is a margin beyond which negative pairs do not contribute to $\mathcal{L}_2$. The adopted metric for the computation of $\mathcal{E}_h$ is the Euclidean distance; additionally, a further loss function $\mathcal{L}_3$ is introduced by exploiting the cosine similarity S_c as a metric for the square-square loss. Specifically:

$$\begin{cases} \mathcal{E}_h = \|\mathbf{h}_1 - \mathbf{h}_2\|_2 & \text{for} \quad \mathcal{L}_1, \mathcal{L}_2 \ , \\ \mathcal{E}_h = 1 - S_c \ , \quad S_c = \frac{\mathbf{h}_1 \cdot \mathbf{h}_2}{\|\mathbf{h}_1\|_2 \|\mathbf{h}_2\|_2} & \text{for} \quad \mathcal{L}_3 \ , \end{cases} \tag{5}$$

where S_c measures the cosine of the angle between $\mathbf{h}_1$ and $\mathbf{h}_2$, so that $S_c = 1$ if $\mathbf{h}_1$ and $\mathbf{h}_2$ are parallel, and $S_c = 0$ if $\mathbf{h}_1$ and $\mathbf{h}_2$ are orthogonal. The reason to consider $\mathcal{L}_3$ as a loss function is that embeddings associated to different conditions are constrained to feature a relative rotation in the low-dimensional space, rather than a different module. This induces an auto-regularization effect that stabilizes the training, and should improve the representation capability of the network.

An interesting perspective to frame the proposed damage localization strategy can be built upon energy-based models (EBMs) [44]. EBMs rely on an energy function $\mathcal{E}(\mathcal{Y}, \mathcal{X})$ that measures the level of compatibility between observed variables $\mathcal{X}$ and the variables to be inferred $\mathcal{Y}$. The learning phase consists in finding, among a family of energy functions $\mathcal{E}(\boldsymbol{\Omega}, \mathcal{Y}, \mathcal{X})$ parametrized by $\boldsymbol{\Omega}$, the one that provides low energy values for a correct association of $\mathcal{Y}$ to $\mathcal{X}$ and high energy values otherwise. The shaping of $\mathcal{E}(\boldsymbol{\Omega}, \mathcal{Y}, \mathcal{X})$ is achieved through the minimization of a loss functional $\mathcal{L}(\mathbf{D}, \mathcal{E}(\boldsymbol{\Omega}, \mathcal{Y}, \mathcal{X}))$ to optimize $\boldsymbol{\Omega}$. The inference of $\mathcal{Y}$ thus consists in simply picking its value that minimizes the learned energy function, given $\mathcal{X}$. Contrastive methods shape the energy function through a loss functional designed to yield: low values for either low energies on correct associations or high energies on incorrect associations; high values for either high energy on correct associations or low energy on incorrect associations. In this work, the energy function of interest is represented by $\mathcal{E}_h$; the mean squared error loss and square-square loss in Eqs. (3) and (4), respectively, are the considered loss functionals. The former implicitly pushes up the energy for all the possible outcomes, while pushing down the energy for the correct association; the latter instead pushes down the energy for the correct association and pushes up the energy above a prescribed margin for incorrect answers.

For SHM purposes, the mapping $G(\cdot)$ is required to be invariant under transformations on the input space not related to the damage position. To this goal, an extractor of damage-sensitive features is implemented at the beginning of the sub-network model, through the composition of three one-dimensional (1D) convolutional units. Convolutional layers are chosen because they naturally embed good relational inductive biases such as locality (the convolutional kernel inputs are in close proximity with one another and isolated from distal entities) and translation equivariance (the ability to reuse the learned rule across different localities in the input) [10, 45], which prove highly effective to analyze signals like multivariate time series. Each convolutional unit consists of a convolutional layer, a nonlinear activation function, batch normalization, a 1D max pooling layer and a dropout layer. The extracted features are expected to be sensitive to the presence of damage, but insensitive to measurement noise and operational variability. The output features are then flattened and

run through a stack of three dense layers, whose output is the low-dimensional representation $\mathbf{h}$. No activation is applied to the last dense layer, while the hyperbolic tangent function is employed as activation function elsewhere. A schematic representation of the subnetwork is reported in Fig. 1. More insights on the working principle of convolutional units and dense layers are reported in Appendix A, for the sake of completeness.

This specific NN architecture, as well as its hyperparameters, are selected through a preliminary sensitivity study, aimed at reducing the computational cost while retaining the damage localization capacity. Details related to the NN architecture and the training options are summarized in Tab. 1. Adopting the Glorot's weights initialization [46], the loss function is minimized using the Adam optimization algorithm [47] for a maximum of 100 epochs. The learning rate η is initially set to $10^{-3}/10^{-4}$, and decreased for 80 epochs by adopting a cosine decay schedule with weight decay 0.05. An early-stopping strategy is used to avoid overfitting. The NN architecture is implemented through the `Tensorflow`-based `Keras` API [48], and trained on a single `Nvidia GeForce RTX 3080` GPU card.

Table 1: selected hyperparameters and training options.

Kernels sizes:	$H_1 = 25 - H_2 = 13 - H_3 = 7$	Weight initializer:	Glorot uniform
Kernel filters:	$N_1 = 16 - N_2 = 32 - N_3 = 16$	Optimizer:	Adam
Depth of dense layers:	$N_4 = 64 - N_5 = 16 - N_6 = D_h$	Batch size:	32
Initial learning rate:	$\eta = \{10^{-3}, 10^{-4}\}$	Allowed epochs:	100
Final learning rate:	0.05 initial value	Early stopping patience:	15 epochs
Learning schedule:	80 epochs cosine decay	Train-validation split:	80 : 20

Once the NN is trained, the damage position for data unseen during the training phase is predicted in the low-dimensional space. To this aim, the Siamese architecture as well as $\mathbf{D}_P$ are discarded, and only the trained subnetwork $G(\cdot)$ together with $\mathbf{D}$ are retained. The damage position is then predicted using the k-nearest neighbors (KNN) algorithm [49, 50] performing a regression on top of mapping $G(\cdot)$, similarly to [51]. Besides its effectiveness, the KNN algorithm is chosen because it is a nonparametric, memory-based method and is therefore extremely fast at both learning and prediction stages, as long as the dimensionality of data is limited.

Denoting by $\mathbf{U}^{(u)}$ the incoming (unknown) vibration recordings and $\mathbf{h}^{(u)} = G(\mathbf{U}^{(u)})$ the correspondent query mapping, the KNN regression algorithm works as follows. During training, the recordings in $\mathbf{D}$ are mapped onto the low-dimensional space, once and for all, to provide

$$\mathbf{D}_h = \{(\mathbf{h}_i, \boldsymbol{y}_i)\}_{i=1}^{I} \ , \tag{6}$$

so that the low-dimensional maps and the relative labels collected in $\mathbf{D}_h$ are stored by the KNN model. At prediction time, the model computes the distances among $\mathbf{h}_i$, $i = 1, \dots, I$, and $\mathbf{h}^{(u)}$ to evaluate a weighted average over its K nearest neighbors, providing an estimate $\hat{\boldsymbol{y}}^{(u)}$ of the associated damage position $\boldsymbol{y}^{(u)}$ according to

$$\hat{\boldsymbol{y}}^{(u)} = \frac{\sum_{k=1}^{K} \chi_k(\mathbf{h}^{(u)}, \mathbf{h}^{(k)}) \boldsymbol{y}^{(k)}}{\sum_{k=1}^{K} \chi_k(\mathbf{h}^{(u)}, \mathbf{h}^{(k)})} \ , \quad k = 1, \dots, K. \tag{7}$$

In Eq. (7), the weighting rule can be uniform, i.e. $\chi_k = 1 \ \forall k$, or based on the inverse distance weights $\chi_k = \frac{1}{\mathcal{E}_h(\mathbf{h}^{(u)}, \mathbf{h}^{(k)})}$, where $\mathbf{h}^{(k)} \in \mathbf{D}_h, k = 1, \dots, K$, is the k-th closest neighbor of

$\mathbf{h}^{(u)}$. The number of neighbors K is not set a-priori; its optimal value and the relative weighting rule are determined through an N-fold cross-validation strategy. This latter works as follows [52]: the training data are randomly split into $\mathfrak{N}$ distinct folds; the model is trained and evaluated $\mathfrak{N}$ times, picking a different evaluation fold every time and leaving all the others for training; for each possible combination of K and weighting rule, a total regression score is obtained by averaging over each validation fold; the model class providing the maximum fold-averaged score is selected and retrained using the full dataset. This approach is expected to be effective in reducing the potential influence of outliers within $\mathbf{D}_h$, as it should identify the model of optimal complexity.

It might be noticed that, fitting the KNN regressor on the reference low-dimensional representations $\mathbf{h}_i$, $i = 1, \dots, I$, also adopted for training $G(\cdot)$, may yield poor generalization capabilities. Indeed, as learning machines best perform on training data, the associated low-dimensional representation may be too optimistic compared to that obtained for data unseen during training. Even if in this work this issue does not show up, since any possibility of overfitting is prevented, it could be beneficial for the damage localization performance to exploit an additional budget of recordings to fit the KNN regressor.

2.3. Numerical modeling of the structural behavior

The structure to be monitored is modeled as an elastic continuum, discretized in space by means of finite elements. Its dynamic behavior, under the assumption of linearized kinematics, is governed by the following semi-discretized form of the elasto-dynamic problem

$$\begin{cases} \mathbf{M}\ddot{\mathbf{d}}(t) + \mathbf{K}(\boldsymbol{\mathcal{y}}, \delta)\mathbf{d}(t) = \mathbf{f}(t, \boldsymbol{\eta}) \ , & t \in (0, T) \\ \mathbf{d}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) = \dot{\mathbf{d}}_0 \ , \end{cases} \tag{8}$$

which is referred to as full-order model (FOM). Here: $t \in (0, T)$ denotes time; $\mathbf{d}(t) \in \mathbb{R}^{\mathscr{M}}$ and $\ddot{\mathbf{d}}(t) \in \mathbb{R}^{\mathscr{M}}$ are the vectors of nodal displacements and accelerations, respectively; $\mathscr{M}$ is the number of degrees of freedom (dofs); $\mathbf{M} \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}}$ is the mass matrix; $\mathbf{K}(\boldsymbol{\mathcal{y}}, \delta) \in \mathbb{R}^{\mathscr{M} \times \mathscr{M}}$ is the stiffness matrix; $\boldsymbol{\mathcal{y}}$ and δ, that are fixed within $(0, T)$, provide the dependence of $\mathbf{K}$ on the damage position and magnitude, respectively; $\mathbf{f}(t, \boldsymbol{\eta}) \in \mathbb{R}^{\mathscr{M}}$ is the vector of nodal forces associated to the operational conditions, parametrized by the vector $\boldsymbol{\eta}$; $\mathbf{d}_0$ and $\dot{\mathbf{d}}_0$ are the initial conditions (at $t = 0$) in terms of nodal displacements and velocities, respectively. Because of the small relevance in the identification of continuously excited systems, according to, e.g. [53, 54], structural damping is disregarded.

Adopting a uniform partition of the time interval $(0, T)$, with a time step size set on the basis of the fundamental structural frequencies and of the sensor sampling rate, the solution of problem (8) is advanced in time using the Newmark time integration scheme [55], to provide the vectors of displacements $\mathbf{d}_l$ and accelerations $\ddot{\mathbf{d}}_l$, with $l = 1, \dots, L$.

As $\mathscr{M}$ increases, the computational cost associated to the solution of the FOM for any sampled $(\boldsymbol{\eta}, \boldsymbol{\mathcal{y}}, \delta)$ grows, and the generation of synthetic datasets becomes prohibitive. Therefore, following the same strategy adopted in [17, 56], the construction of $\mathbf{D}$ is speeded up through the use of a projection-based ROM strategy for parametrized systems. The RB method [35], relying on the proper orthogonal decomposition (POD)-Galerkin approach, is here adopted because of its appealing offline-online decoupling. Moreover, POD has been

largely investigated in the context of structural dynamics [57–59], so that clear algebraic and geometric interpretations of the POD-basis functions are available, together with efficient criteria for their selection.

The solution of problem (8) is thus approximated, in terms of displacements, as $\mathbf{d}(t,\boldsymbol{\eta},\boldsymbol{\mathcal{y}},\delta) \approx \mathbf{W}\mathbf{d}_R(t,\boldsymbol{\eta},\boldsymbol{\mathcal{y}},\delta)$ through a linear combination of $\mathfrak{R} \ll \mathscr{M}$ basis functions $\mathbf{w}_\mathfrak{r} \in \mathbb{R}^{\mathscr{M}}$, $\mathfrak{r}=1,\dots,\mathfrak{R}$, gathered in the matrix $\mathbf{W}=[\mathbf{w}_1,\dots,\mathbf{w}_\mathfrak{R}] \in \mathbb{R}^{\mathscr{M}\times\mathfrak{R}}$.

$\mathbf{W}$ is obtained by performing a singular value decomposition (SVD) of a snapshot matrix $\mathbf{S}=[\mathbf{d}_1,\dots,\mathbf{d}_S] \in \mathbb{R}^{\mathscr{M}\times S}$. Matrix $\mathbf{S}$ collects S snapshots of the FOM, namely solutions in terms of time histories of nodal displacements, obtained by solving problem (8) for different values of the parameters $(\boldsymbol{\eta},\boldsymbol{\mathcal{y}},\delta)$ selected through the latin hypercube sampling. The SVD factorizes matrix $\mathbf{S}$ as

$$\mathbf{S}=\mathbf{P}\boldsymbol{\Sigma}\mathbf{Z}^\top \ , \tag{9}$$

where: $\mathbf{P}=[\mathbf{p}_1,\dots,\mathbf{p}_\mathscr{M}] \in \mathbb{R}^{\mathscr{M}\times\mathscr{M}}$ is an orthogonal matrix, whose columns are the left singular vectors of $\mathbf{S}$; $\boldsymbol{\Sigma} \in \mathbb{R}^{\mathscr{M}\times S}$ is a pseudo-diagonal matrix collecting the singular values σ_l, $l=1,\dots,\mathfrak{P}$, of $\mathbf{S}$, arranged so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_\mathfrak{P} \geq 0$, being $\mathfrak{P}=\min(S,\mathscr{M})$ the rank of $\mathbf{S}$; $\mathbf{Z}=[\mathbf{z}_1,\dots,\mathbf{z}_S] \in \mathbb{R}^{S\times S}$ is another orthogonal matrix, whose columns are the right singular vectors of $\mathbf{S}$.

According to the Schmidt-Eckart-Young theorem, see [35, 60], the first $\mathfrak{R}$ columns of $\mathbf{P}$ minimize the reconstruction error, when approximating $\mathbf{S}$ by a matrix of rank $\mathfrak{R}$. The ROM order $\mathfrak{R}$ is then obtained by prescribing the error in energy to be smaller than a given tolerance ϵ, according to:

$$\frac{\sum_{l=1}^{\mathfrak{R}}(\sigma_l)^2}{\sum_{l=1}^{\mathfrak{P}}(\sigma_l)^2} \geq 1-\epsilon \ . \tag{10}$$

For any parameter set not accounted for in the construction of the ROM itself, the reduced-order approximation to the solution of problem (8) is determined by solving the following $\mathfrak{R}$-dimensional problem:

$$\begin{cases} \mathbf{M}_R\ddot{\mathbf{d}}_R(t)+\mathbf{K}_R(\boldsymbol{\mathcal{y}},\delta)\mathbf{d}_R(t)=\mathbf{f}_R(t,\boldsymbol{\eta}) \ , \quad t\in(0,T) \\ \mathbf{d}_R(0)=\mathbf{W}^\top\mathbf{d}_0 \\ \dot{\mathbf{d}}_R(0)=\mathbf{W}^\top\dot{\mathbf{d}}_0 \ , \end{cases} \tag{11}$$

resulting by enforcing the orthogonality between the residual $\mathbf{f}(t,\boldsymbol{\eta})-\mathbf{M}\mathbf{W}\ddot{\mathbf{d}}_R(t)-\mathbf{K}(\boldsymbol{\mathcal{y}},\delta)\mathbf{W}\mathbf{d}_R(t)$ and the subspace $\text{span}\{\mathbf{w}_1,\dots,\mathbf{w}_\mathfrak{R}\}$, where:

$$\mathbf{M}_R \equiv \mathbf{W}^\top\mathbf{M}\mathbf{W} \ , \qquad \mathbf{K}_R(\boldsymbol{\mathcal{y}},\delta) \equiv \mathbf{W}^\top\mathbf{K}(\boldsymbol{\mathcal{y}},\delta)\mathbf{W} \ , \qquad \mathbf{f}_R(t,\boldsymbol{\eta}) \equiv \mathbf{W}^\top\mathbf{f}(t,\boldsymbol{\eta}). \tag{12}$$

The approximated solution is then recovered by back-projecting the ROM solution whenever necessary, via $\mathbf{d}(t) \approx \mathbf{W}\mathbf{d}_R(t)$, or $\ddot{\mathbf{d}}(t) \approx \mathbf{W}\ddot{\mathbf{d}}_R(t)$ depending on the handled measurements. Nodal displacements or accelerations in $(0,T)$ are collected as $\mathbf{V}=[\mathbf{d}_1,\dots,\mathbf{d}_L] \in \mathbb{R}^{\mathscr{M}\times L}$ or $\mathbf{V}=[\ddot{\mathbf{d}}_1,\dots,\ddot{\mathbf{d}}_L] \in \mathbb{R}^{\mathscr{M}\times L}$, so that the sensor recordings $\mathbf{U}$ are ultimately obtained as

$$\mathbf{U}=(\mathbf{T}\mathbf{V})^\top \ , \tag{13}$$

where $\mathbf{T} \in \mathbb{B}^{N_u\times\mathscr{M}}$ is a Boolean matrix whose (n,m)-th entry is equal to 1 only if the m-th dof coincide with the n-th sensor output.

The algorithmic description of the offline phase of the proposed methodology is reported in Algorithm 1. During the next online phase, new vibration recordings provided by the sensing system are mapped onto the discriminative low-dimensional feature space and the position of the damage, if any, is predicted.

Algorithm 1 Offline phase of the proposed methodology: algorithmic description

Set up the Physics-based numerical model of the structure to be monitored
    Parametrize the relevant operational (and possibly environmental) conditions
    Parametrize the relevant damage conditions

Build the ROM according to the POD approach of the RB method
    Run FOM analyses to generate snapshots of the structural response:
        Sample the input parameters
        Run the FOM simulation and store snapshots of the solution in time
    Perform the SVD to find the POD-basis functions

Generate a synthetic dataset through the ROM
    Run ROM analyses to simulate the structural response:
        Sample the input parameters
        Run the ROM simulation
        Extract the nodal dofs mimicking the relevant sensing system
        Store the vibration recordings and the associated label

Train the damage localization model
    Augment the dataset through the pairing process
    Train the feature extractor by means of the Siamese network
    Map labeled data onto the discriminative low-dimensional space
    Fit the prediction strategy on the low-dimensional features

In the present work, the FOM and the ROM are implemented in the `Matlab` environment, using the `redbKIT` library [61]. All computations are carried out on a desktop PC featuring an AMD Ryzen$^{\text{TM}}$, 9 5950X CPU @ 3.4 GHz and 128 GB RAM.

3. Damage detection in an L-shaped cantilever beam

The first case study adopted to assess the performance of the proposed methodology is the three-dimensional model of the L-shaped cantilever beam depicted in Fig. 2. The structure is composed of two arms, each one having a length of 4 m, a width of 0.3 m and a height of 0.4 m. The model is discretized using linear tetrahedral elements, with the entire mesh resulting in $\mathscr{M} = 4659$ dofs. The structure is assumed to be made of concrete, whose mechanical properties are: Young's modulus $E = 30$ GPa, Poisson's ratio $\nu = 0.2$, density $\rho = 2500$ kg/m^3. The structure is perfectly clamped at one end and is excited close to the other one by a distributed load $q(t)$, acting on an area of (0.3×0.3) m^2, see Fig. 2. The load $q(t)$ varies in time according to $q(t) = Q \sin(2\pi f t)$, where $Q \in [1, 5]$ kPa and $f \in [10, 100]$ Hz respectively denote the load amplitude and frequency, and are collected in $\boldsymbol{\eta} = \{Q, f\}^\top$; the above notation specifying the range of variation of parameters, implicitly assumes that a uniform probability distribution is adopted to describe them.

This first case study aims at verifying the damage localization capability of the proposed method, also illustrating the interpretability of the obtained results. A further purpose is the

identification of the combination of loss function and size D_h of the low-dimensional space that provides the most satisfactory results.

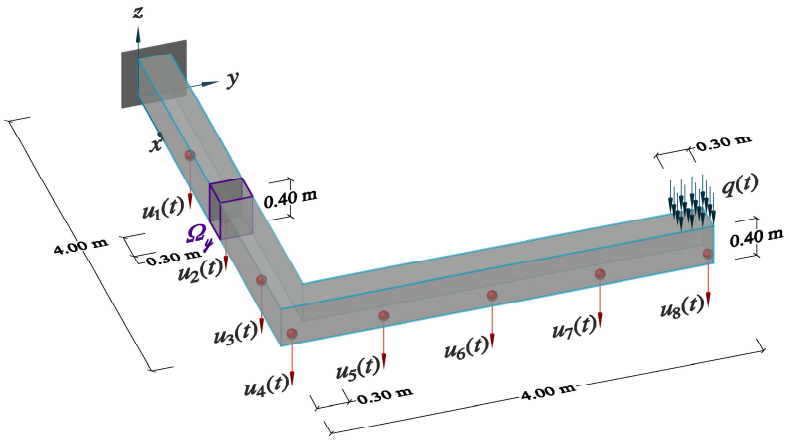


Figure 2: L-shaped cantilever beam: physics-based digital twin, with details of the loading conditions and synthetic recordings related to displacements $u_1(t), \dots, u_8(t)$.

3.1. Dataset assembly and training options

The dataset $\mathbf{D}$, according to Eq. (1), is populated of multivariate time series $\mathbf{U}(\boldsymbol{\eta}, \boldsymbol{y}, \delta)$ obtained by reducing the stiffness within the subdomain $\Omega_{\boldsymbol{y}}$, which is a box $(0.3 \times 0.3 \times 0.4)$ m^3 in size. The target position of this reduction is given by the coordinates of its center of mass $\boldsymbol{y} = (x_\Omega, y_\Omega) \in \mathbb{R}$, with $x_\Omega, y_\Omega \in [0.15, 3.85]$ m, which vary one at a time and can be encoded as an abscissa running along the axis of the structure. The damage level $\delta \in [10\%, 25\%]$ is held constant within the time instance $(0, T)$.

Synthetic displacement recordings $\mathbf{U}(\boldsymbol{\eta}, \boldsymbol{y}, \delta) = [\mathbf{u}_1, \dots, \mathbf{u}_{N_u}]$ are obtained in relation to $N_u = 8$ dofs along the bottom surface of the structure, to mimic a monitoring system arranged as depicted in Fig. 2. Each recording $\mathbf{u}_n(t)$, $n = 1, \dots, N_u$, is provided with a frequency of acquisition of $\mathscr{f} = 500$ Hz over the time interval $(0, T = 1$ s$)$, thus consisting of $L = 501$ data points. Measurement noise, represented by an additive white Gaussian noise featuring a signal-to-noise ratio of 80, is then used to corrupt the recordings.

The ROM is built exploiting a snapshot matrix $\mathbf{S}$ featuring $S = 150,000$ snapshots, obtained through 300 evaluations of problem (8) carried out over $(0, T)$, at varying values of the input parameters. By prescribing a tolerance $\epsilon = 0.5 \cdot 10^{-3}$, $\mathfrak{R} = 68$ POD bases are selected in place of the original 4659 dofs; thanks to the ROM, the computing time required by each simulation decreases from 11.30 s to 0.39 s, entailing a speed-up of about 29 times. For a deeper discussion concerning these aspects in a DL-integrated SHM framework see [17].

For the case at hand, $I = 8000$ instances are collected using the ROM. The augmented dataset $\mathbf{D}_P$ in Eq. (2) is constructed pairing each instance with $\tau_P = 10$ similar instances yielding $\mathcal{E}_{\boldsymbol{y}} \leq \overline{\mathcal{E}_{\boldsymbol{y}}}$ and with $\tau_N = 10$ dissimilar instances yielding $\mathcal{E}_{\boldsymbol{y}} > \overline{\mathcal{E}_{\boldsymbol{y}}}$, after prescribing $\overline{\mathcal{E}_{\boldsymbol{y}}} = 0.15$ m on the basis of the aforementioned size of the damaged box. The testing data consist of $I_T = 800$ pseudo-experimental instances, generated instead through the noise-corrupted FOM solutions.

3.2. Damage localization outcomes

The results regarding the achieved performance are now commented considering the adoption of the loss functions $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ introduced in Sec. 2.2. A comparison among the three different outcomes is finally reported and discussed.

3.2.1. MSE loss function

The damage localization performance obtained by adopting at the training stage the MSE loss function $\mathcal{L}_1$, equipped with an Euclidean metric, is first considered. In this case, the size $D_h = 2$ of the low-dimensional space directly follows from the modeling of the considered problem, without any need of tuning. The low-dimensional mapping obtained for the training data is reported in Fig. 3a, where $\{(h_1, h_2)_i\}_{i=1}^{I}$ are plotted together with a color channel referring to the target damage position along the x-axis, i.e. $\{x_{\Omega i}\}_{i=1}^{I}$, and along the y-axis, i.e. $\{y_{\Omega i}\}_{i=1}^{I}$, see also Fig. 2. The overall shape defined by the scatter plots reflects the actual structural geometry, and the colors are in agreement with the coordinates in the physical space. Hence, the mapping is properly learned and the low-dimensional features provide a system of coordinates suitable to catch the damage position from the vibration recordings.

The testing phase attempts to mimic a real monitoring process of a structural system: raw vibration recordings are therefore assumed to be collected on-the-fly, and analyzed to localize the damage position. The mapping of the testing data is reported in Fig. 3b, to show that the NN has a proper generalization capacity. Damage is then located in real-time by exploiting the KNN algorithm described in Sec. 2.2. $K = 58$ nearest neighbors and the uniform weighting rule are adopted through a 20-fold cross-validation, as providing the maximum folds-averaged regression score.

The distributions of the prediction error along the x and y directions, related to the testing data, are reported in Fig. 4. In these charts, the counts refer to the number of instances for which a certain prediction error is reported. The observed values are distributed in a rather narrow range around the origin, showing that the damage position is almost always identified with high accuracy. It can be noted that the error distribution along the y-direction features a slightly greater variance with respect to what happens along the x-direction; this is due to the structural layout, which yields a sensitivity of sensor recordings to damage smaller for damage positions far from the clamped side.

Assuming the prediction errors along the same two directions to follow a Gaussian distribution, some related statistics are also reported in Tab. 2. In this table, the mean average error (MAE), the error mean μ and standard deviation ζ, along the x and y directions, are employed to assess the performance of the damage localization model. In particular, the MAE values have to be highlighted, which are smaller than half the size of the damaged region in both directions; the μ values are close to zero, as it should be for an unbiased estimator; the ζ values confirm the greater inaccuracies along the y-direction, as well as the MAE values. Accordingly, 95% of the testing population is reported to yield prediction errors lying within the intervals $[-0.38, 0.43]$ m and $[-0.43, 0.51]$ m, respectively along the x and y directions, which is a remarkable result.

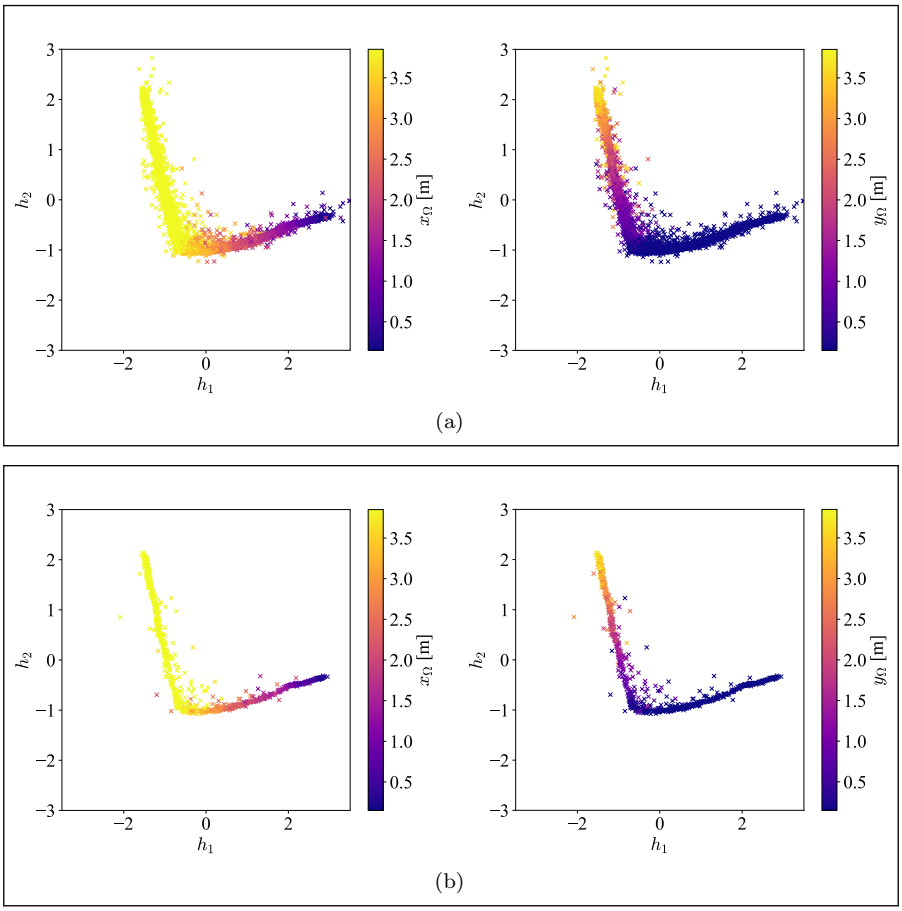


Figure 3: L-shaped cantilever beam - mapping learned via $\mathcal{L}_1$ and $D_h = 2$: (a) training instances $\{(h_1, h_2)_i\}_{i=1}^{I}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2)_i\}_{i=1}^{I_T}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I_T}$.

Table 2: L-shaped cantilever beam - testing data, accuracy of damage localization along the x and y directions for $\mathcal{L}_1$ and $D_h = 2$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

I_T:	800	k:	58 – uniform
MAE_x:	0.094 m	MAE_y:	0.117 m
μ_x:	0.028 m	μ_y:	0.043 m
ζ_x:	0.21 m	ζ_y:	0.24 m

3.2.2. Square-square loss function with Euclidian metric

The results obtained using the square-square loss function $\mathcal{L}_2$, equipped with an Euclidean metric, are now described. By minimizing $\mathcal{L}_2$, dissimilar pairs are kept away by at least the margin ψ, while similar pairs are pushed to be as close as possible. The mapping is not explicitly constrained by the label values as in the previous case, and the NN is left free to learn a suitable metric space by itself. This aspect results in the need of increasing the size D_h of the low-dimensional mapping, to provide a sufficient representation capability to properly describe the input variability to the NN. In order to identify a suitable criterion

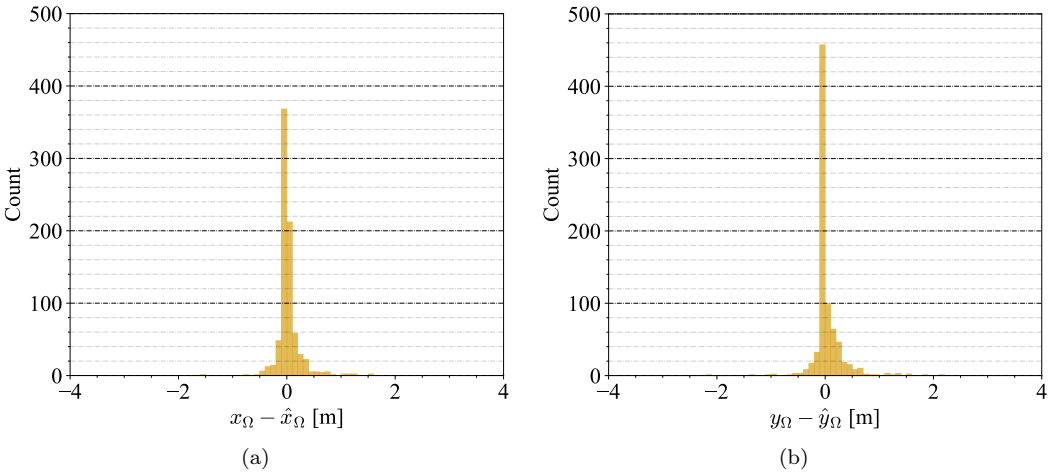


Figure 4: L-shaped cantilever beam - distribution of the prediction error over the testing instances along the (a) x-direction and (b) y-direction, adopting $\mathcal{L}_1$ and $D_h = 2$.

to select D_h, the performances of the NN are systematically analyzed for $D_h = 2 - 4$. By slightly anticipating the discussion of the results, the best outcome is obtained with $D_h = 4$, with no improvement for $D_h > 4$. This is somehow expected, being $D_h = 4$ equal to the number of parameters, i.e. the intrinsic dimension of the structural model.

A compact, two-dimensional representation of the embedding for the training and testing data related to $D_h = 4$ is reported in Fig. 5, as obtained by applying the multidimensional scaling (MDS) to the NN mappings. MDS is a nonlinear dimensionality reduction method in statistical learning that seeks a low-dimensional representation of the input data, still preserving the pairwise scalar products [62, 63]; the metric implementation in `scikit-learn` [64] is here adopted. As expected, the overall shape defined by the scatter plots does not resemble the actual structural layout in this case, as the relevant geometrical information are not exploited during the training. Graphs showing the general low-dimensional mapping for the training and testing data are instead reported in Fig. 6 and Fig. 7, respectively. Here, $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I,I_T}$ are plotted in pairs, against the correct damage position along the x and y directions, providing the low-dimensional manifold described by the learned mapping as the position of damage varies. From both the MDS representation and the general mapping it can be claimed that the task-specific mapping is learned with a proper generalization capacity, as consistent results are obtained for the training and testing data.

The distributions of the prediction error along the x and y directions relative to the testing data are reported in Fig. 8, only for $D_h = 4$. The two distributions confirm that the damage position is almost always identified with high accuracy, yet less accurately along the y-direction, according to the results obtained in Sec. 3.2.1. The statistics of the prediction errors for the considered choices of the embedding size D_h are shown in Tab. 3. The performance of the damage localization model is more than satisfactory even for $D_h = 2$, although increasing D_h yields an improvement of all the considered metrics, as the NN features a greater representation capability useful to describe the variability of the structural response properly. In all cases, the reported MAE values are smaller than, or equal to,

half the size of the damaged region in both directions and the best result is obtained with $D_h = 4$, improving the result provided with $\mathcal{L}_1$ in all the considered performance measures. Adopting this setting, and according to the reported mean and standard deviation values, 95% of the testing population yields prediction errors bounded within $[-0.36, 0.42]$ m and $[-0.44, 0.54]$ m, respectively along the x and y directions. In spite of the improvement reported for higher values of D_h, the computational time required to train and evaluate the NN is not affected. Indeed, choosing $D_h = 4$ has a negligible impact on the number of tunable parameters $\boldsymbol{\Omega}$, yet the relevant representation capability of the NN significantly improves. The comparison among the results obtained with $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ is postponed to Sec. 3.3.

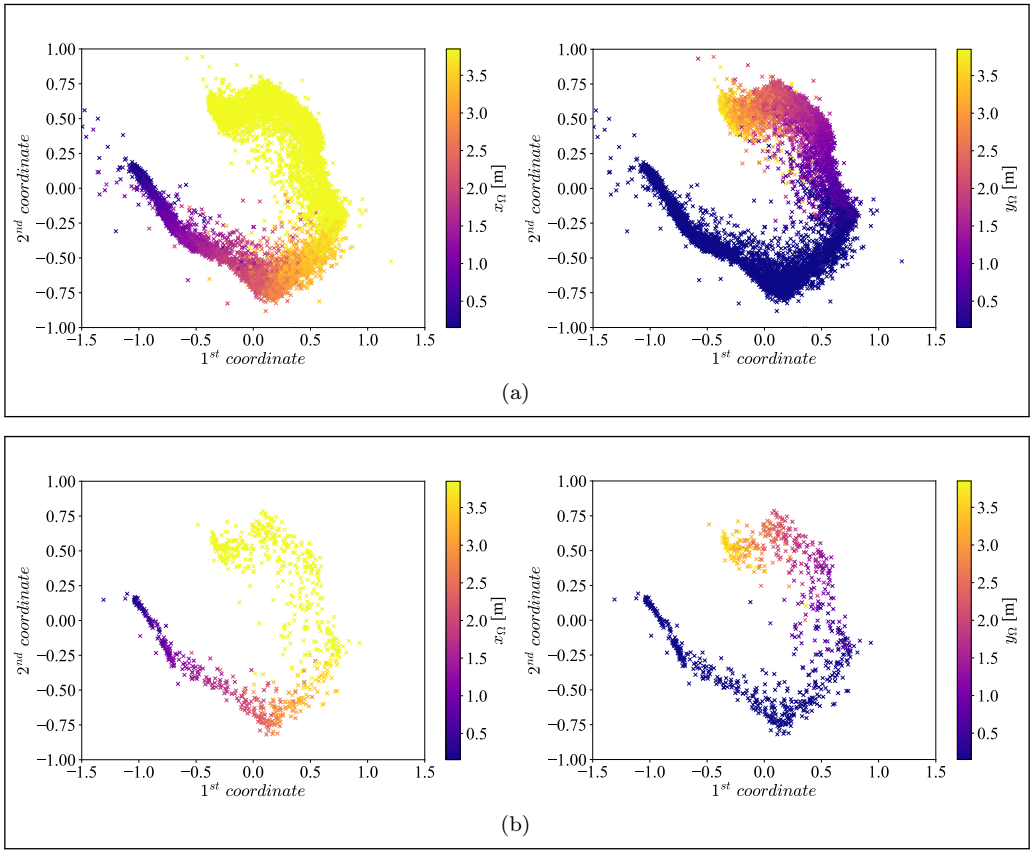


Figure 5: L-shaped cantilever beam - multidimensional scaling onto two dimensions of the mapping learned via $\mathcal{L}_2$ and $D_h = 4$: (a) training instances $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I}$, against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I_T}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I_T}$.

3.2.3. Square-square loss function with cosine similarity

The results obtained with the square-square loss function $\mathcal{L}_3$, equipped with the metric built upon the cosine similarity defined in Eq. (5), are now detailed. In this case, the performances of the NN are systematically analyzed in the range $D_h = 3 - 5$. As compared to what reported for $\mathcal{L}_2$, the size of the low-dimensional space is increased, to account for

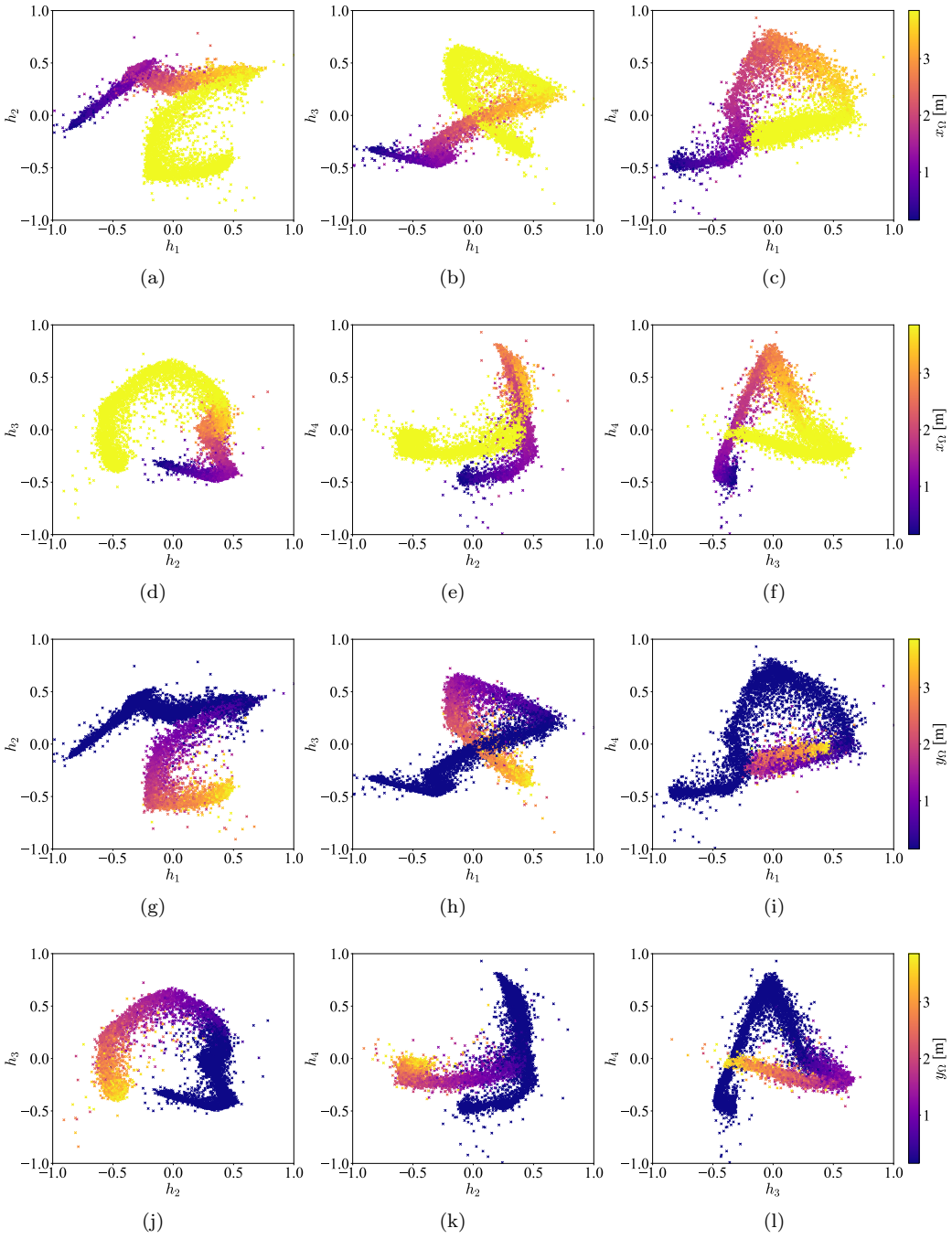


Figure 6: L-shaped cantilever beam - mapping of the training data $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^I$ plotted 2-by-2 against the correct damage position using $\mathcal{L}_2$ and $D_h = 4$, along the (a-f) x-direction $\{x_{\Omega i}\}_{i=1}^I$ and (g-l) y-direction $\{y_{\Omega i}\}_{i=1}^I$.

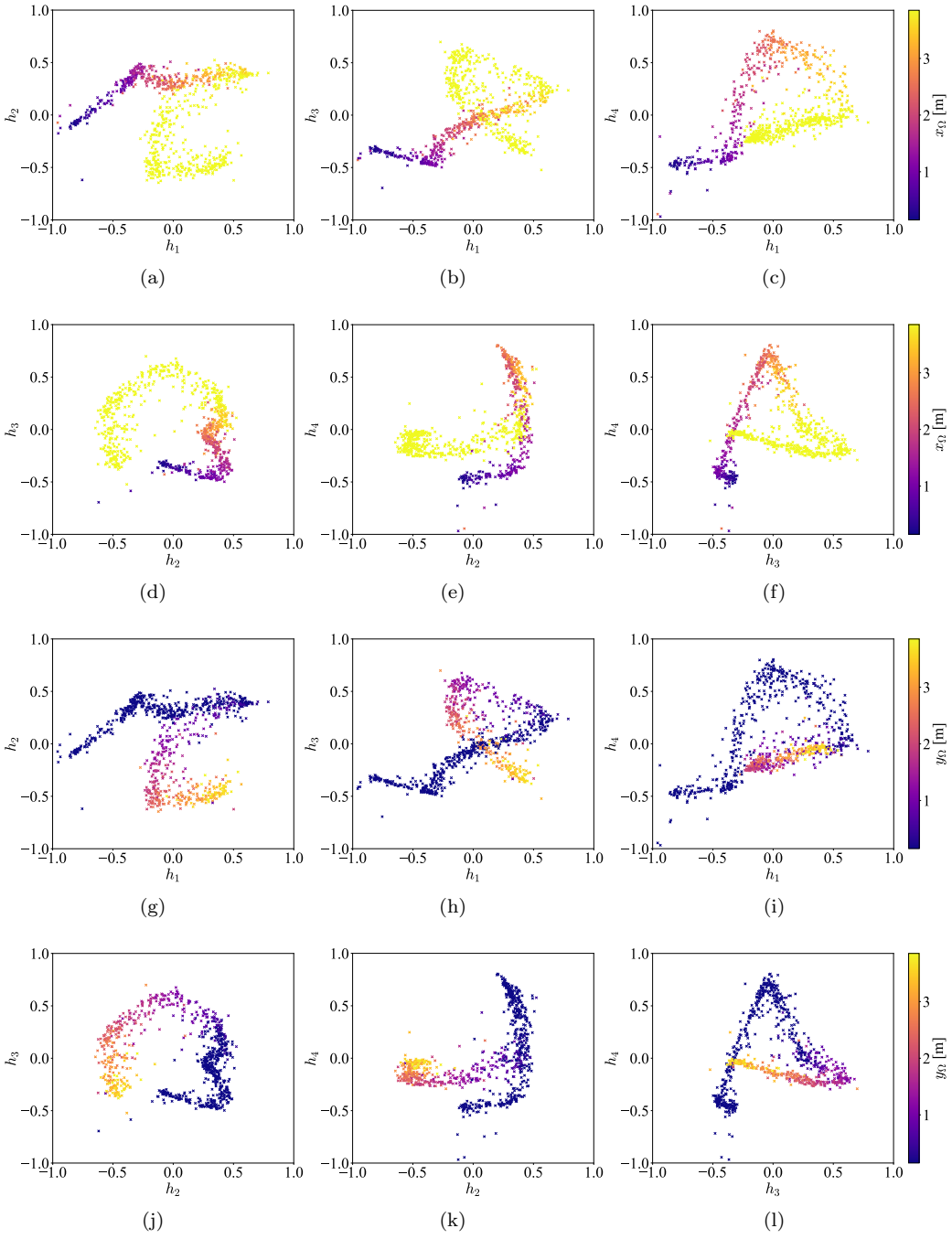


Figure 7: L-shaped cantilever beam - mapping of the testing data $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I_T}$ plotted 2-by-2 against the correct damage position using $\mathcal{L}_2$ and $D_h = 4$, along the (a-f) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (g-l) y-direction $\{y_{\Omega i}\}_{i=1}^{I_T}$.

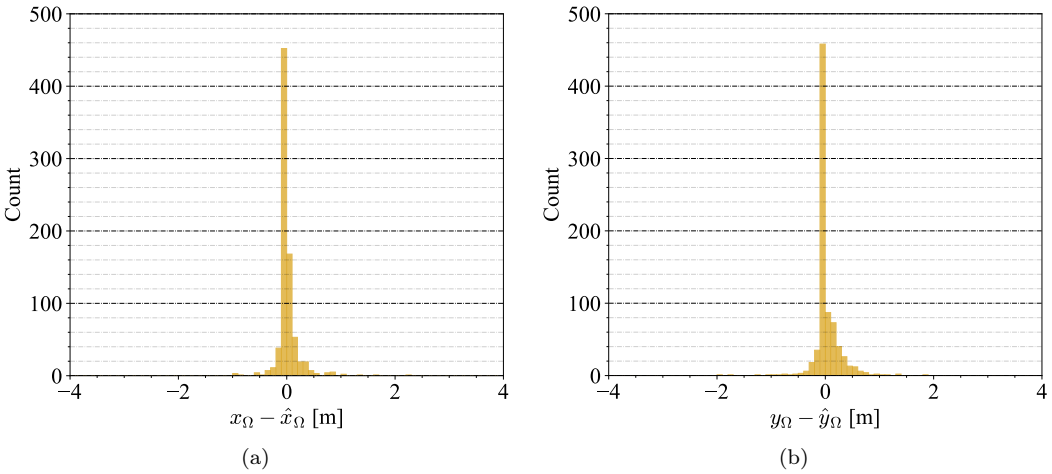


Figure 8: L-shaped cantilever beam - distribution of the prediction error over the testing instances along the (a) x-direction and (b) y-direction, adopting $\mathcal{L}_2$ and $D_h = 4$.

Table 3: L-shaped cantilever beam - testing data, accuracy of damage localization along the x and y directions for $\mathcal{L}_2$ and $D_h = \{2, 3, 4\}$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

$I_T = 800$	$D_h = 2$	$D_h = 3$	$D_h = 4$
k:	96 − unif.	56 − dist.	51 − dist.
MAE_x:	0.120 m	0.091 m	0.082 m
MAE_y:	0.152 m	0.116 m	0.105 m
μ_x:	0.037 m	0.038 m	0.028 m
μ_y:	0.073 m	0.078 m	0.047 m
ζ_x:	0.26 m	0.21 m	0.20 m
ζ_y:	0.30 m	0.27 m	0.25 m

the normalization constraint provided by S_c. Anticipating the discussion of the results, and according to what obtained with $\mathcal{L}_2$, the best results are provided by $D_h = 5$.

The MDS representations onto two dimensions of the embedding obtained with the training and testing data for $D_h = 5$ are reported in Fig. 9. It is interesting to note how the manifold described by the learning mapping increases in size for damage positions far from the clamped side, because of their smaller impact on the sensor recordings yielding a less accurate identification. For completeness, the embedding on the unit hypersphere obtained from the general low-dimensional mapping is reported in Fig. 10, as obtained for $D_h = 3$ only; it should be remarked that $D_h = 3$ is here considered in place of $D_h = 5$, to fully display an exemplary metric space built upon the cosine similarity. It can be noticed that embeddings associated with different damage conditions are forced to differentiate themselves by featuring a rotation in the low-dimensional space.

The distributions of the prediction error along the x and y directions in relation to the testing data are reported in Fig. 11, only for $D_h = 5$. It has to be noted that the same metric built upon the cosine similarity used during the training, has to be adopted also for the computation of distances at prediction stage. The observed values are distributed similarly to what reported for $\mathcal{L}_1$ and $\mathcal{L}_2$, but it can be qualitatively appreciated that damage

localizations are slightly less accurate for $\mathcal{L}_3$. The statistics of the prediction errors obtained at varying size D_h of the low-dimensional space are shown in Tab. 4. In this case, increasing D_h enables only a partial enhancement: the performance measures along the y-direction are not improving for higher D_h values, while those along the x-direction improve only by a limited amount in comparison to what reported for $\mathcal{L}_2$. As anticipated, the best result is reported for the largest size $D_h = 5$ of the low-dimensional space among those considered: 95% of the testing population yields prediction errors bounded within $[-0.48, 0.54]$ m and $[-0.47, 0.59]$ m, respectively along the x and y directions.

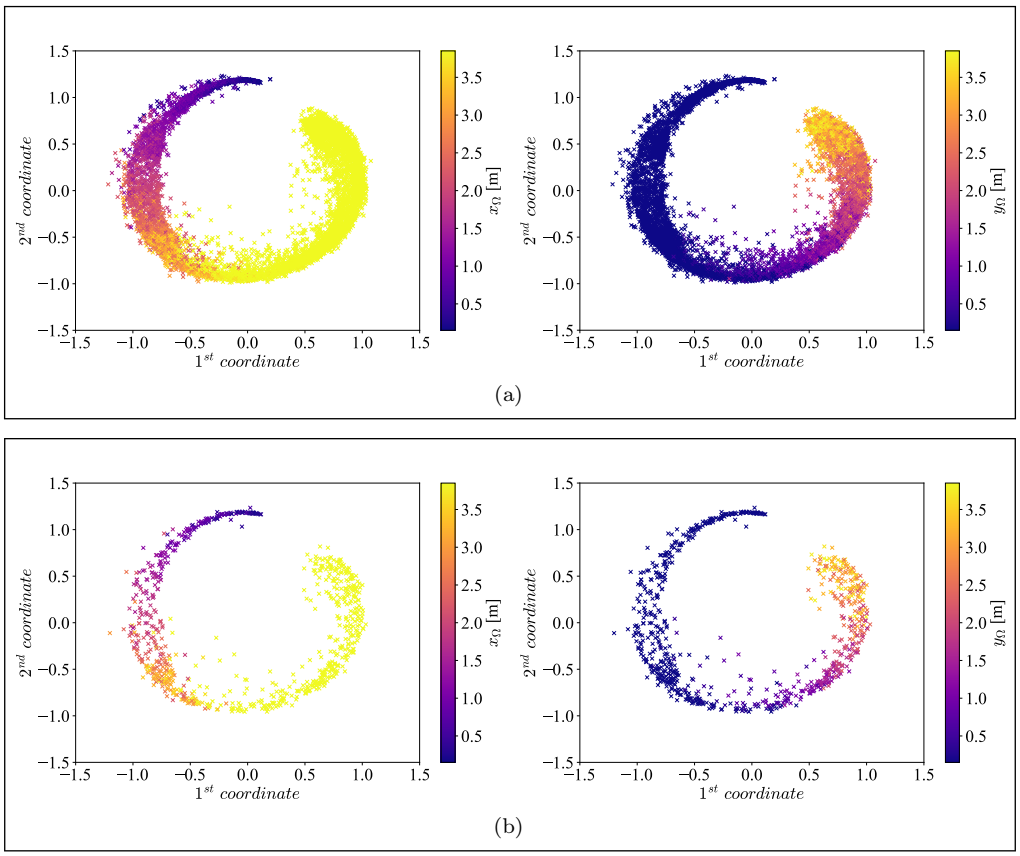


Figure 9: L-shaped cantilever beam - multidimensional scaling onto two dimensions of the mapping learned via $\mathcal{L}_3$ and $D_h = 5$: (a) training instances $\{(h_1, h_2, h_3, h_4, h_5)_i\}_{i=1}^{I}$, against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2, h_3, h_4, h_5)_i\}_{i=1}^{I_T}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I_T}$.

3.3. Discussion of the results

The results obtained for the present case study are in general fairly good. Given the structural layout, the sensitivity of measurements to damage is smaller for damage positions far from the clamped side; accordingly, the NN performs better along the x-direction. For all considered sizes of the low-dimensional space, the three loss functions yield MAE values smaller than, or equal to, half the size of the damaged region in both directions, which is a remarkable result. The worst localization outcomes are obtained with the NN leveraging

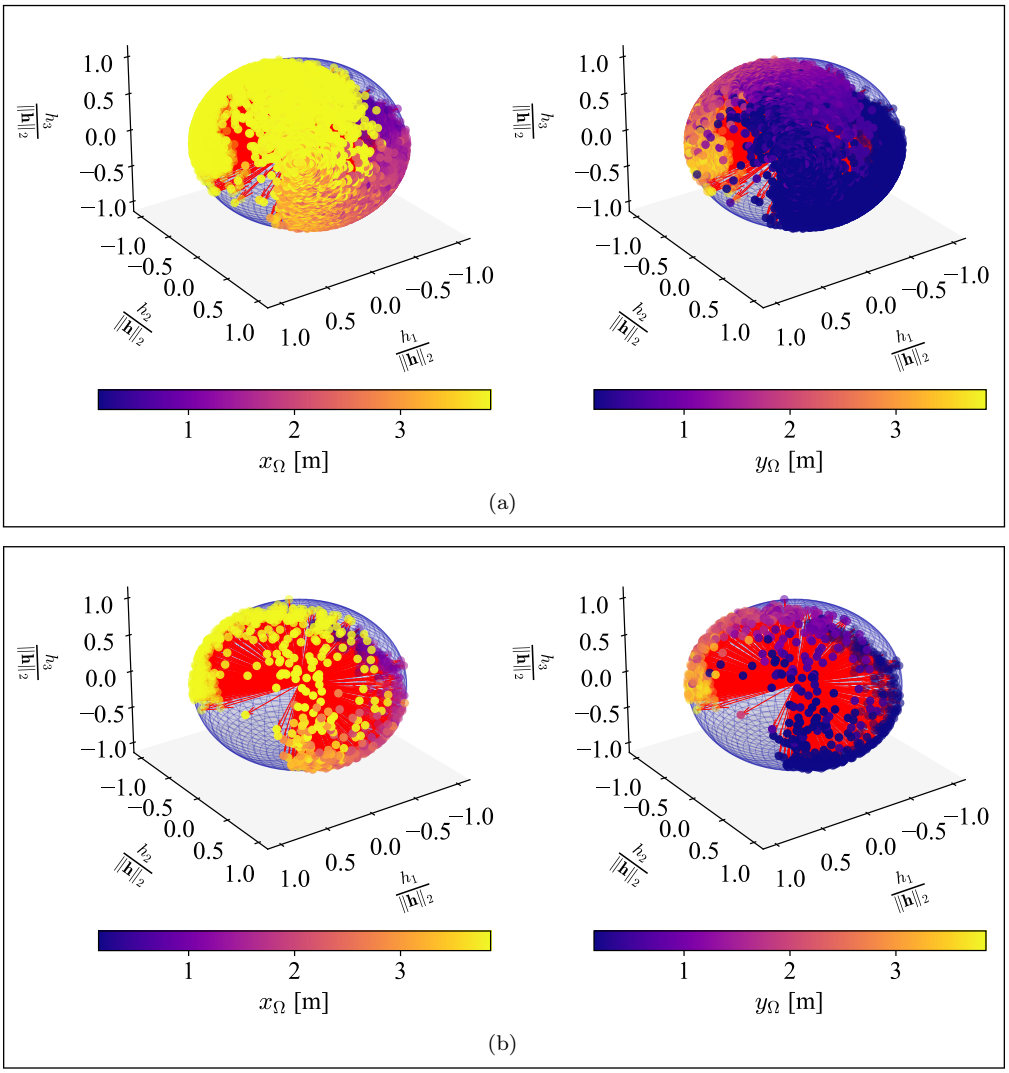
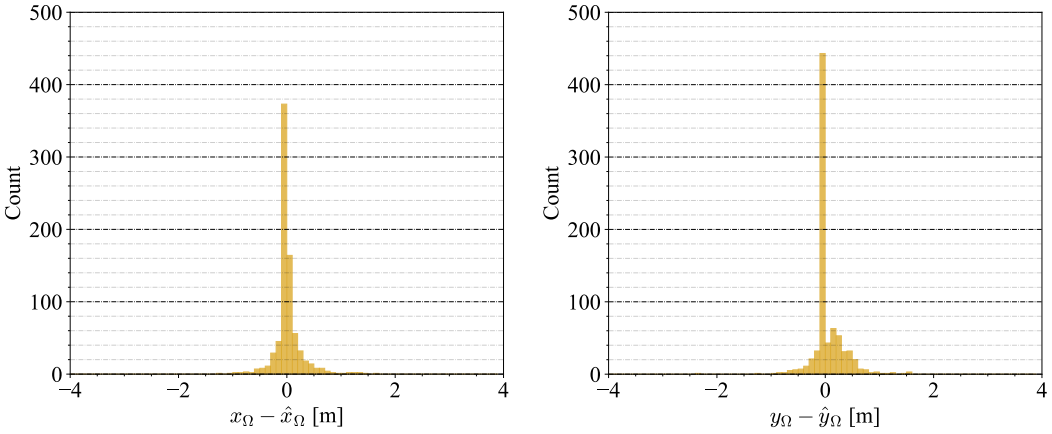


Figure 10: L-shaped cantilever beam - normalized embedding of the mapping learned via $\mathcal{L}_3$ and $D_h = 3$: (a) training instances $\{(h_1, h_2, h_3)_i\}_{i=1}^{I}$, against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2, h_3)_i\}_{i=1}^{I_T}$, against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) y-direction $\{y_{\Omega i}\}_{i=1}^{I_T}$.

Table 4: L-shaped cantilever beam - testing data, accuracy of damage localization along the x and y directions for $\mathcal{L}_3$ and $D_h = \{3, 4, 5\}$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

$I_T = 800$	$D_h = 3$	$D_h = 4$	$D_h = 5$
k:	120 – unif.	103 – unif.	82 – dist.
MAE_x:	0.143 m	0.132 m	0.125 m
MAE_y:	0.148 m	0.146 m	0.148 m
μ_x:	0.032 m	0.038 m	0.032 m
μ_y:	0.074 m	0.075 m	0.061 m
ζ_x:	0.27 m	0.26 m	0.26 m
ζ_y:	0.27 m	0.27 m	0.27 m



(a)

Figure 11: L-shaped cantilever beam - distribution of the prediction error over the testing instances along the (a) x-direction and (b) y-direction, adopting $\mathcal{L}_3$ and $D_h = 5$.

$\mathcal{L}_3$ during training, which looks almost insensitive to the selected dimension D_h of the low-dimensional space. On the other side, the most accurate results are obtained with the NN leveraging $\mathcal{L}_2$ during training. In this latter case, increasing D_h provides a non-negligible improvement of the localization outcomes. The positive effect of increasing D_h can also be appreciated by the inverse distance weighting rule selected by the cross-validation method in place of the uniform weighting rule, which intuitively testifies the goodness of the learned mapping. The NN trained on $\mathcal{L}_1$ provides instead slightly worse localization results, while the associated low-dimensional mapping is easier to interpret. In light of these considerations, only the loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ are accounted for in the reminder of the research activity here reported.

To assess the robustness of the proposed methodology against the uncertainties related to the size of the damaged region, the setting yielding the most accurate localization outcomes, leveraging the $\mathcal{L}_2$ loss function, is also tested on a further dataset made of $I_T = 800$ pseudo-experimental instances. These instances are generated with the FOM and noise-corrupted as before, but the size of the subdomain wherein the stiffness is reduced by $\delta \in [10\%, 25\%]$ is also varied in its longitudinal thickness l_Ω, according to $l_\Omega \in [0.2, 0.45]$ m. Such variation is accounted for in the testing dataset, while the same value $l_\Omega = 0.3$ m of the previous case is used for training. The obtained results are summarized in Tab. 5, in terms of the statistics of the prediction errors. The obtained outcomes are very close to those reported in Tab. 3: the MAE values are smaller than half the size of the damaged region in both directions; the prediction error mean values are close to zero along the x and y directions; according to the slightly higher deviations than that observed in the reference testing condition, 95% of the testing population yields prediction errors bounded within $[-0.54, 0.63]$ m and $[-0.56, 0.66]$ m, respectively along the x and y directions, which is a remarkable result considering damaged regions of variable size and damage level.

Table 5: L-shaped cantilever beam - testing data characterized by damaged regions of varying size, accuracy of damage localization along the x and y directions for $\mathcal{L}_2$ and $D_h = 4$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

I_T:	800	k:	51 – dist.
MAE_x:	0.122 m	MAE_y:	0.133 m
μ_x:	0.045 m	μ_y:	0.048 m
ζ_x:	0.30 m	ζ_y:	0.31 m

4. Damage detection in a railway bridge

This second case study aims to assess the proposed methodology in a more complex situation, involving the integral concrete portal frame railway bridge depicted in Fig. 12. The bridge is located along the Bothnia line in the urban area of Hörnefors in the northern Sweden. It features a span of 15.7 m, a free height of 4.7 m and a width of 5.9 m (edge beams excluded). The wing walls stretch out in the longitudinal direction, up to 8 m at the top. The thickness of the structural elements is 0.5 m for the deck, 0.7 m for the frame walls, and 0.8 m for the wing walls. The bridge is founded on a couple of plates connected by two stay beams and supported by pile groups. The superstructure consists of a single ballasted track, 0.6 m deep and 4.3 m wide, that rests on sleepers spaced 0.65 m apart. All the geometrical and mechanical data used in this experiment are adapted from former research activities involving the analysis of the relevant soil-structure interaction, see [65, 66].

Figure 12: Hörnefors railway bridge.

The bridge is subjected to the transit of trains type Gröna Tåget, at a speed $v \in [160, 215]$ km/h. Only trains composed of two wagons are considered, thus characterized by 8 axles, see also [17], each one carrying a mass $\phi \in [16, 22]$ ton. The equivalent distributed force is transmitted to the structure through 25 sleepers over the deck, and then propagated through the ballast with a slope $4:1$, according to Eurocode 1 [67]. Denoting by p_{max} the maximum load that each axle can provide to the structure, the entire moving load system is modeled as:

$$P(t, v, x) = \sum_{\iota=1}^{25} \sum_{\kappa=1}^{8} p_{max} \cdot \mathcal{A}^{\iota}(x) \cdot \mathcal{A}^{\iota}_{\kappa}(t, v) \ , \tag{14}$$

with sums of all contributions due to the 8 axles and the 25 sleepers. Herein, $\mathcal{A}^{\iota}(x)$ is the space activation function of the ι-th sleeper, and $\mathcal{A}^{\iota}_{\kappa}(t, v)$ is the relevant time modulation

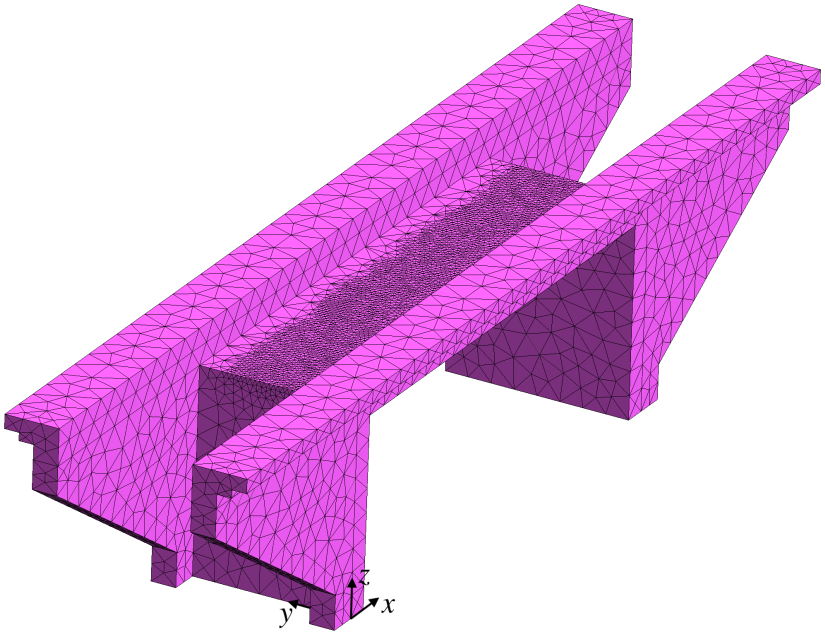


Figure 13: FE discretization of the Hörnefors railway bridge.

function related to the κ-th axle; further details on $\mathcal{A}^{\iota}(x)$ and $\mathcal{A}^{\iota}_{\kappa}(t, v)$ can be found in [17].

The bridge is discretized with $17,057$ linear tetrahedral elements as shown in Fig. 13, resulting in $\mathscr{M} = 17,292$ dofs. To ensure a smooth transmission of the moving load, the space discretization features a reduced element size of 0.15 m for the deck, while it is set to 0.80 m elsewhere. The concrete is assumed of class C35/45, whose mechanical properties are: $E = 34$ GPa, $\nu = 0.2$, $\rho = 2500$ kg/m^3. The ballast, whose density is $\rho = 1800$ kg/m^3, is modeled through an increased density of the deck and of the edge beams. The presence of the embankments is accounted for through a Robin mixed boundary condition (with elastic coefficient $k_{\text{Robin}} = 10^8$ N/m^3) over the surface facing the ground. The dynamic response of the bridge to the train passage is computed within the time interval $(0, T = 1.5$ s) with a time step size of $1/\mathscr{f}$, $\mathscr{f} = 400$ Hz, in order to both allow the train to cross the bridge even when traveling at the lowest speed and properly catch the structural effects at the maximum speed of 215 km/h.

4.1. Dataset assembly and training options

$\mathbf{D}$ is assembled by considering that damage can take place anywhere over the two lateral frame walls and the deck, within subdomains $\Omega_{\boldsymbol{y}}$ featuring a different layout in the two cases, see Fig. 14. The localized stiffness reduction can occur with magnitude $\delta \in [30\%, 40\%]$ fixed within $(0, T)$, and its target coordinates are given by $\boldsymbol{y} = (x_\Omega, z_\Omega) \in \mathbb{R}$, with $x_\Omega \in [-0.115, 16.63]$ m and $z_\Omega \in [0.25, 6.25]$ m varying one at a time.

The monitoring system is devised to provide synthetic displacement recordings $\mathbf{U}(\boldsymbol{\eta}, \boldsymbol{y}, \delta) = [\mathbf{u}_1, \dots, \mathbf{u}_{N_u}]$, related to the $N_u = 10$ dofs highlighted in Fig. 14. Each time history $\mathbf{u}_n(t)$, $n = 1, \dots, N_u$, collects $L = 601$ data points. To allow for the sensor noise, all the data are corrupted with an independent, identically distributed Gaussian noise, featuring a signal-to-noise

ratio equal to 120.

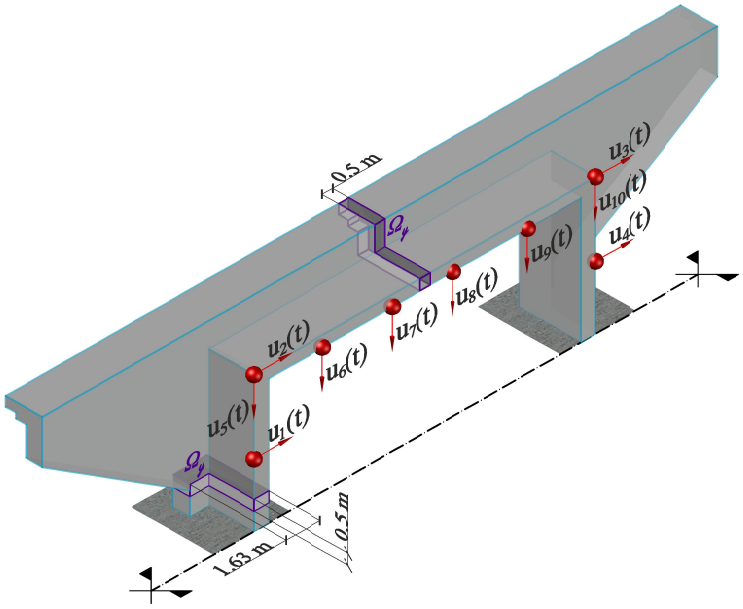


Figure 14: Railway bridge - physics-based digital twin, perspective cross-section with details of the damaged region Ω_y and synthetic recordings related to displacements $u_1(t), \ldots, u_{10}(t)$.

The number of snapshots used to construct the ROM is set to $S = 180,000$, resulting from 300 FOM analyses, for different input parameters. To build the ROM, the error tolerance is set to $\epsilon = 10^{-3}$, yielding $\mathfrak{R} = 528$ in place of the original $17,292$ dofs; the CPU time required by each simulation thus decreases from 82.95 s to 12.50 s, yielding a speed-up of about 6.90 times. This slightly large number of POD-basis functions is mainly due to the need of embedding into the ROM the specific response of the structure, in the presence of a variety of potential localized stiffness reductions.

The dataset $\mathbf{D}$ is built with $I = 8000$ instances computed with the ROM, which are augmented to $\mathbf{D}_P$ with $\tau_P = \tau_N = 10$ pairs after setting $\overline{\mathcal{E}_y} = 0.25$ m. The testing dataset consists of $I_T = 800$ pseudo-experimental instances, generated through the FOM and noise corrupted. Time histories of the vertical displacement at midspan are reported in Fig. 15, to compare the FOM and ROM solutions in the absence of the added measurement noise, to assess the ROM accuracy, and, in the presence of noise, to also assess its potential impact on the handled structural response to the loading.

4.2. Damage localization Outcomes

In view of the results reported for the L-shaped cantilever beam case, only the loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$ are considered for the present investigation.

4.2.1. MSE loss function

If the MSE loss function $\mathcal{L}_1$ equipped with an Euclidean metric is adopted, the size of the low-dimensional mapping is set to $D_h = 2$. The low-dimensional mappings obtained with the training and testing data are reported in Figs. 16a and 16b, respectively. The representation

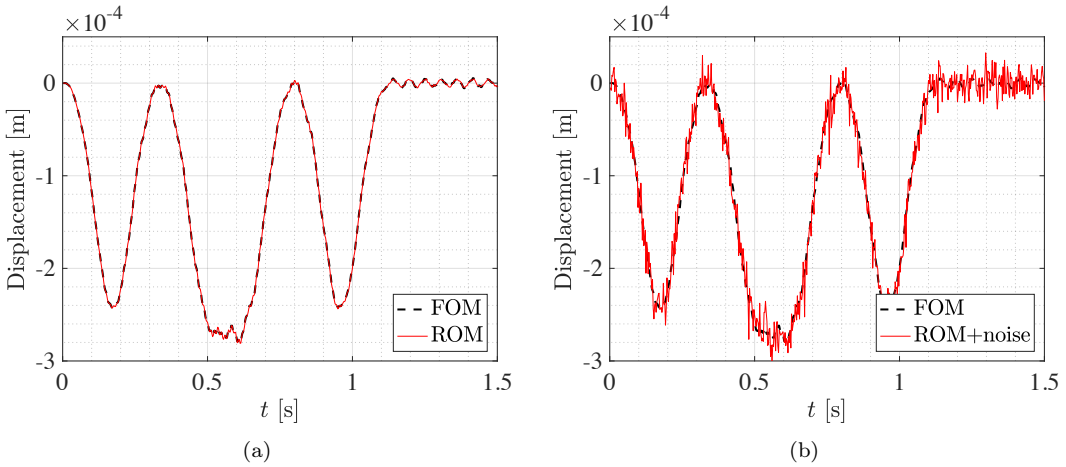


Figure 15: Railway bridge - vertical displacement time histories at midspan ($x_\Omega = 8.25$ m, $z_\Omega = 6.25$ m, $\delta = 0.35$, $v = 210$ km/h and $\phi = 20$ ton): comparison between (a) FOM and reduced-order approximation, and (b) FOM and noise-corrupted reduced-order approximation.

of the two components $\{(h_1, h_2)_i\}_{i=1}^I$, against the position of damage along the x and z directions, proves that the mapping is properly learned, as the overall shape defined by the scatter plots clearly reflects the geometrical parametrization of the damage position over the bridge, described by the color channel. It can also be claimed that the NN is not incurred in overfitting issues, as the low-dimensional representations obtained with the training and testing data show an extremely good agreement.

Regarding the KNN regressor, $K = 42$ nearest neighbors and the uniform weighting rule are selected by the cross-validation method, to predict the damage position. The distributions of the prediction error along the x and z directions relative to the testing data are reported in Fig. 17. The two distributions provide a first qualitative indication that the damage position is almost always identified with great accuracy, as the observed values are concentrated in a rather narrow range around the origin. The associated statistics are shown in Tab. 6, reporting drastically larger inaccuracies in locating damage along the x-direction: both the MAE and the standard deviation values are more than doubled with respect to the z-direction. Despite that, the damage position is often provided with a high level accuracy: assuming the observed prediction errors to be Gaussian distributed, 95% of the testing population yields prediction errors lying within the intervals $[-2.54, 2.63]$ m and $[-1.16, 1.19]$ m, respectively along the x and z directions.

Table 6: Railway bridge - testing data, accuracy of damage localization along the x and z directions for $\mathcal{L}_1$ and $D_h = 2$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

I_T:	800	k:	42 – unif.
MAE_x:	0.692 m	MAE_z:	0.306 m
μ_x:	0.043 m	μ_z:	0.019 m
ζ_x:	1.32 m	ζ_z:	0.60 m

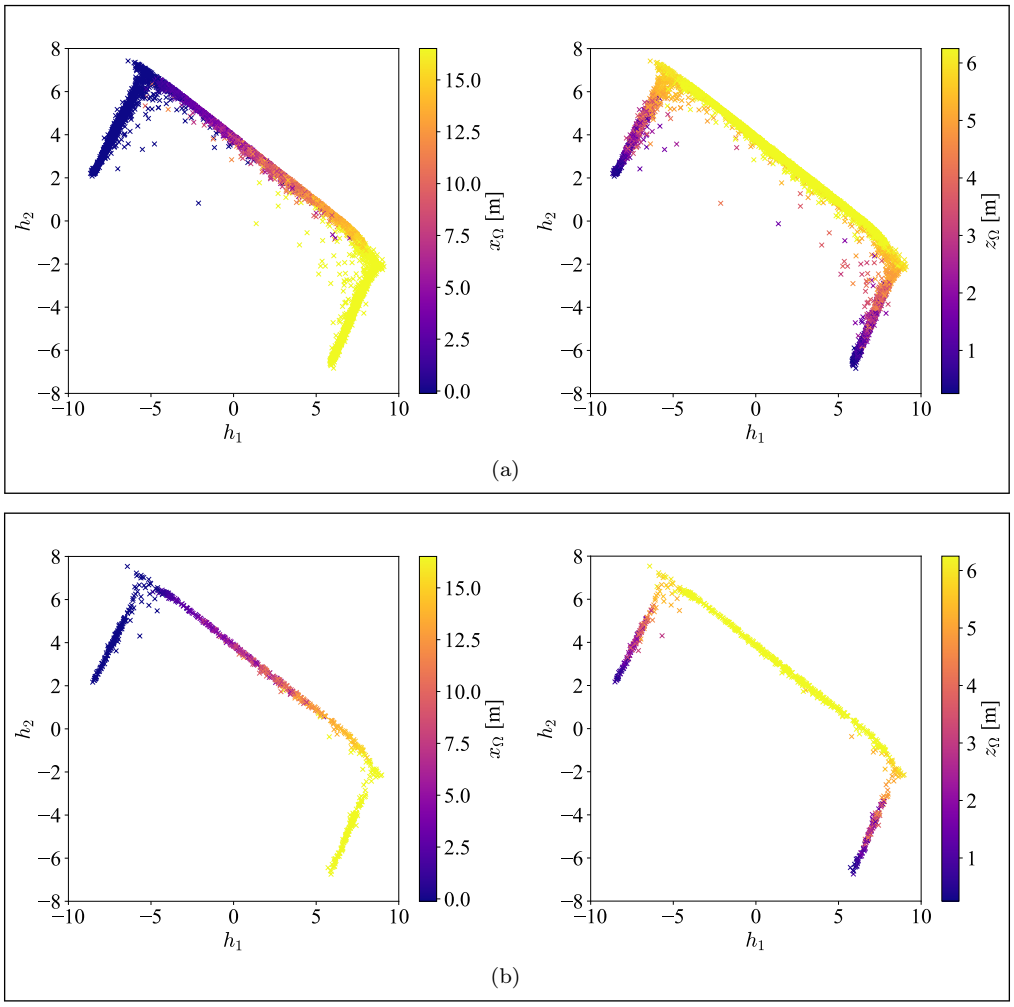


Figure 16: Railway bridge - mapping learned via $\mathcal{L}_1$ and $D_h = 2$: (a) training instances $\{(h_1, h_2)_i\}_{i=1}^{I}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) z-direction $\{z_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2)_i\}_{i=1}^{I_T}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) z-direction $\{z_{\Omega i}\}_{i=1}^{I_T}$.

4.2.2. Square-square loss function with Euclidian metric

On the basis of the results presented in Sec. 3.2, the square-square loss function $\mathcal{L}_2$ equipped with an Euclidean metric is adopted by choosing the size of the low-dimensional mapping to be $D_h = 4$, thus equal to the number of free parameters ruling the structural model. The MDS representations onto two dimensions, obtained with the training and testing data, are reported in Figs. 18a and 18b, respectively. These plots qualitatively show that the damage position is encoded in the low-dimensional space with higher accuracy along the z-axis, as the portion of the manifold describing the variation along the x-axis looks more confused and increased in size, which is in accordance to the results reported for $\mathcal{L}_1$.

$K = 51$ nearest neighbors and the inverse distance weighting rule are adopted through the cross-validation method, to set the KNN regressor. The damage localization outcomes are reported in Fig. 19, in terms of the distributions of prediction error along the x and z

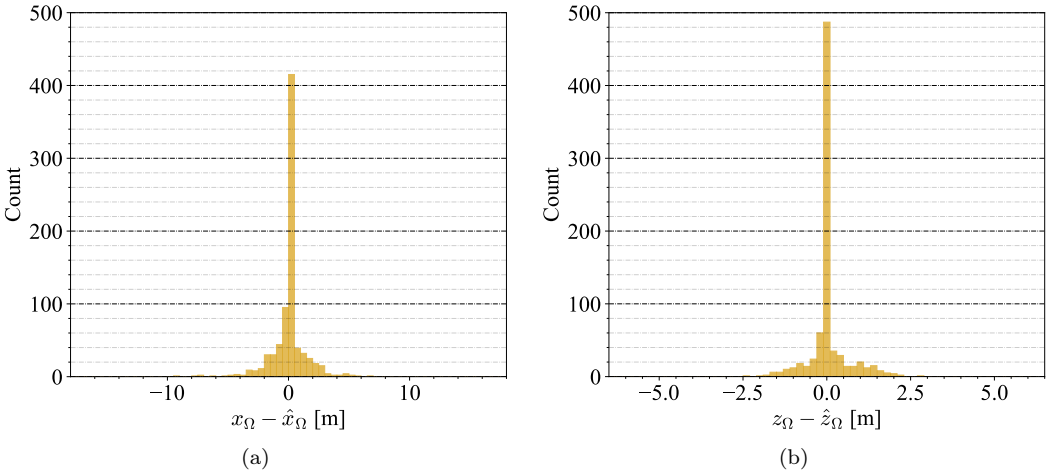


Figure 17: Railway bridge - distribution of the prediction error over the testing instances along the (a) x-direction and (b) z-direction, adopting $\mathcal{L}_1$ and $D_h = 2$.

directions. It can be qualitatively appreciated that the two distributions feature a smaller variance with respect to those obtained leveraging $\mathcal{L}_1$. Such claim is confirmed by the relative statistics reported in Tab. 7, that show an improvement over all the considered performance metrics, in particular along the x-direction: the MAE values improve from 0.692 m and 0.306 m to 0.453 m and 0.260 m, respectively along the x and z directions; 95% of the testing population yields prediction errors lying within the intervals $[-2.07, 2.12]$ m and $[-0.95, 1.01]$ m along the same directions. This further validates the square-square loss function $\mathcal{L}_2$ equipped with an Euclidean metric as the most satisfactory one.

According to the obtained results, the proposed methodology is shown to outperform the former damage detection strategy, proposed by the authors in [17]. The latter is based upon a supervised DL-based classification framework, exploited to locate damage in structures by mapping raw vibration recordings onto discrete target labels, associated to predefined damage scenarios of interest. Considering the very same railway bridge case study, the quality of the outcomes is improved in several ways: the level of accuracy at which the damage is located is markedly refined, as only six damage scenarios, associated to quite large damaged regions, were accounted in the previous work; during this former investigation, the damage level δ was held constant within all instances, while a range of interest to sample δ is prescribed in the present work; moving from a classification framework toward a metric-based one, it is not required to define a-priori a bunch of target classes, yet the damage can be identified anywhere in the structure.

Table 7: Railway bridge - testing data, accuracy of damage localization along the x and z directions for $\mathcal{L}_2$ and $D_h = 4$. Data reported in terms of: mean absolute error (MAE); mean (μ) and standard deviation (ζ) associated to the error distributions.

I_T:	800	k:	51 – dist.
MAE_x:	0.453 m	MAE_z:	0.260 m
μ_x:	0.028 m	μ_z:	0.027 m
ζ_x:	1.07 m	ζ_z:	0.50 m

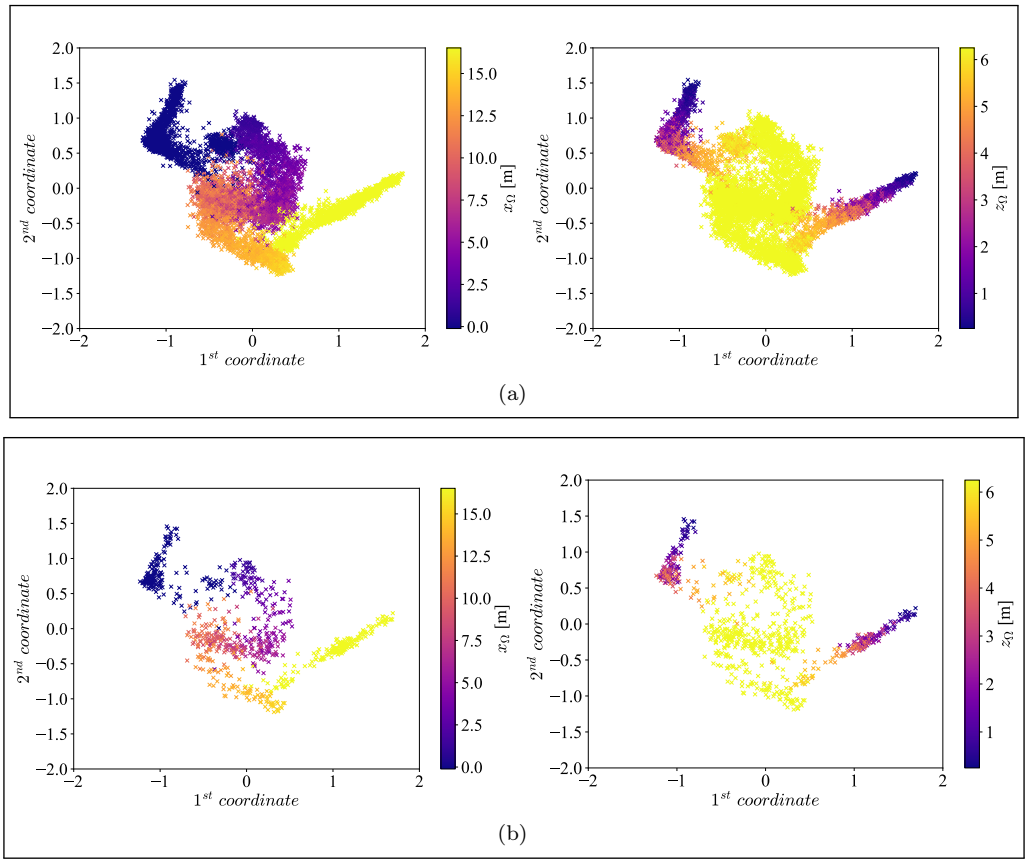


Figure 18: Railway bridge - multidimensional scaling onto two dimensions of the mapping learned via $\mathcal{L}_2$ and $D_h = 4$: (a) training instances $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I}$, against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I}$ and (right) z-direction $\{z_{\Omega i}\}_{i=1}^{I}$; (b) testing instances $\{(h_1, h_2, h_3, h_4)_i\}_{i=1}^{I_T}$ against the correct damage position along the (left) x-direction $\{x_{\Omega i}\}_{i=1}^{I_T}$ and (right) z-direction $\{z_{\Omega i}\}_{i=1}^{I_T}$.

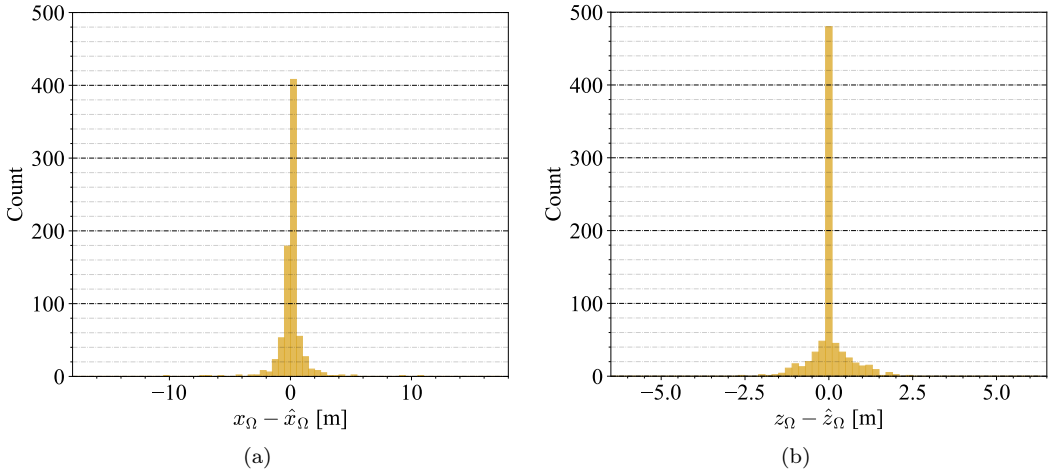


Figure 19: Railway bridge - distribution of the prediction error over the testing instances along the (a) x-direction and (b) z-direction, adopting $\mathcal{L}_2$ and $D_h = 4$.

5. Conclusion

Within a structural health monitoring framework, this paper has developed a new method for damage localization, in case of low evolution rate. The procedure has exploited an ordered and smooth mapping of vibration data onto a low-dimensional space, whose components encode the damage position in the structure.

The training stage has been performed in a supervised pairwise fashion, exploiting synthetic datasets generated through a reduced-order model to speed up the entire procedure and allow to move toward real-time applications. The location of the damaged zone has been carried out through a k-nearest neighbors regression performed on the learned low-dimensional space. Thanks to the link between the internal characterization of the problem and the physical space, the proposed method can be exploited to locate damage at different scales, without the need to define several target damage classes.

The proposed strategy has been assessed on an L-shaped cantilever beam and a portal frame railway bridge, exploring the use of three different loss functions and also varying the size of the low-dimensional feature space. In all the considered cases, as long as the low-dimensional mapping features enough representation capability to describe the variability of the structural response, the damage localization task has been carried out with a remarkable accuracy. The best results have been achieved by using the square-square loss function equipped with an Euclidean metric, to train the neural network used to parametrize the mentioned low-dimensional mapping. The method has shown to be insensitive to measurement noise, to the varying operational conditions characterized by time-varying and moving loads, and to the relevant damage severity. Moreover, it has been also reported to be rather insensitive to the assumed size of the damaged region – an aspect that has been considered to assess the possible effect of an improper setting of the SHM procedure.

Besides the need of validating the proposed diagnostic framework within a suitable experimental framework, the future work will be devoted to also provide a quantification of damage severity, possibly even when material and geometrical nonlinearities simultaneously affect the structural response. An optimal sensor placement strategy, to maximize the sensitivity to damage of the low-dimensional features, will be also envisaged to improve the overall effectiveness of the methodology.

Acknowledgments: The authors are grateful to Luca Rosafalco for several insightful discussions about this research. This work is supported in part by the interdisciplinary Ph.D. Grant "Physics-Informed Deep Learning for Structural Health Monitoring" at Politecnico di Milano.

The authors declare no conflict of interest.

References

[1] S. Arangio and F. Bontempi, "Structural health monitoring of a cable-stayed bridge with Bayesian neural networks," *Struct Infrastruct Eng*, vol. 11, pp. 575–587, 2015.

[2] P. Thoft-Christensen, "Life-cycle cost-benefit analysis of bridges from a user and social point of view," *Struct Infrastruct Eng*, vol. 5, pp. 49–57, 2009.

[3] C. Jones, T. Lethco, J. Curley, and D. Zackin, "A preventable crisis: The economic and human costs of a hudson river rail tunnel shutdown," *Regional Plan Association*, 2019.

[4] L. A. Bull, K. Worden, and N. Dervilis, "Towards semi-supervised and probabilistic classification in structural health monitoring," *Mech Syst Signal Process*, vol. 140, p. 106653, 2020.

[5] S. Doebling, C. Farrar, M. Prime, and D. Shevitz, "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review," *Shock Vib Digest*, vol. 30, p. 249299, 1996.

[6] C. Farrar and K. Worden, *Structural Health Monitoring A Machine Learning Perspective*. Wiley, 2013.

[7] F. Yuan, S. Zargar, Q. Chen, and S. Wang, "Machine learning for structural health monitoring: challenges and opportunities," in *Proceeding of SPIE Smart Structures/NDE 2020*, vol. 11379, p. 1137903, 2020.

[8] O. Fink, Q. Wang, M. Svensen, P. Dersin, W. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Eng Appl Artif Intell*, vol. 92, p. 103678, 2020.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[12] Z. Wang and Y. J. Cha, "Automated damage-sensitive feature extraction using unsupervised convolutional neural networks," in *Proceeding of SPIE Smart Structures/NDE 2018*, vol. 10598, p. 105981J, 2018.

[13] C. Bigoni and J. S. Hesthaven, "Simulation-based anomaly detection and damage localization: an application to structural health monitoring," *Comput Methods Appl Mech Eng*, vol. 363, p. 112896, 2020.

[14] Y. J. Cha and Z. Wang, "Unsupervised novelty detection-based structural damage localization using a density peaks-based fast clustering algorithm," *Struct Health Monit*, vol. 17, pp. 313–324, 2017.

[15] A. Entezami, H. Shariatmadar, and S. Mariani, "Fast unsupervised learning methods for structural health monitoring with large vibration data from dense sensor networks," *Struct Health Monit*, vol. 19, no. 6, pp. 1685–1710, 2020.

[16] T. Taddei, J. D. Penn, M. Yano, and A. T. Patera, "Simulation-based classification; a model-order-reduction approach for structural health monitoring," *Arch Comput Methods Eng*, vol. 25, pp. 23–45, 2018.

[17] L. Rosafalco, M. Torzoni, A. Manzoni, S. Mariani, and A. Corigliano, "Online structural health monitoring by model order reduction and deep learning algorithms," *Comput Struct*, vol. 255, p. 106604, 2021.

[18] N. Gulgec, M. Takáč, and S. Pakzad, "Convolutional neural network approach for robust structural damage detection and localization," *J Comput Civ Eng*, vol. 33, p. 04019005, 2019.

[19] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "Fully convolutional networks for structural health monitoring through multivariate time series classification," *Adv Model Simul Eng Sci*, vol. 7, p. 38, 2020.

[20] M. G. Kapteyn, D. J. Knezevic, and K. Willcox, "Toward predictive digital twins via component-based reduced-order models and interpretable machine learning," in *AIAA Scitech 2020 Forum*, 2020.

[21] G. Mariniello, T. Pastore, C. Menna, P. Festa, and D. Asprone, "Structural damage detection and localization using decision tree ensemble and vibration data," *Comput-Aided Civil Infrastruct Eng*, vol. 36, pp. 1129–1149, 2020.

[22] K. Agathos, S. P. Bordas, and E. Chatzi, "Parametrized reduced order modeling for cracked solids," *Int J Numer Methods Eng*, vol. 121, no. 20, pp. 4537–4565, 2020.

[23] K. Agathos, K. E. Tatsis, K. Vlachas, and E. Chatzi, "Parametric reduced order models for output-only vibration-based crack detection in shell structures," *Mech Syst Signal Process*, vol. 162, p. 108051, 2022.

[24] O. Abdeljaber, O. Avci, S. Kiranyaz, M. Gabbouj, and D. Inman, "Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks," *J Sound Vib*, vol. 388, pp. 154–170, 2017.

[25] T. Zhang, S. Biswal, and Y. Wang, "SHMnet: Condition assessment of bolted connection with beyond human-level performance," *Struct Health Monit*, vol. 19, no. 4, pp. 1188–1201, 2019.

[26] V. Ewald, R. M. Groves, and R. Benedictus, "DeepSHM: a deep learning approach for structural health monitoring based on guided Lamb wave technique," in *Proceeding of SPIE Smart Structures/NDE 2019*, vol. 10970, p. 109700H, 2019.

[27] X. Pan and T. Yang, "Postdisaster image-based damage detection and repair cost estimation of reinforced concrete buildings using dual convolutional neural networks," *Comput-Aided Civil Infrastruct Eng*, vol. 35, no. 5, pp. 495–510, 2020.

[28] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. Inman, "A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications," *Mech Syst Signal Process*, vol. 147, p. 107077, 2021.

[29] M. Kaya and H. Bilge, "Deep metric learning: A survey," *Symmetry*, vol. 11, p. 1066, 2019.

[30] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," 2013. `https://arxiv.org/pdf/1306.6709`.

[31] F. Cakir, K. He, X. Xia, B. Kulis, and S. Sclaroff, "Deep metric learning to rank," in *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, pp. 1861–1870, 2019.

[32] J. Liu, S. Chen, M. Berges, J. Bielak, J. Garrett, J. Kovacevic, and H. Noh, "Diagnosis algorithms for indirect structural health monitoring of a bridge model via dimensionality reduction," *Mech Syst Signal Process*, vol. 136, p. 106454, 2020.

[33] R. Hadsell, S. Chopra, and Y. Lecun, "Dimensionality reduction by learning an invariant mapping," in *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, pp. 1735 – 1742, 2006.

[34] J. Bromley, I. Guyon, Y. Lecun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *Int J Pattern Recognit Artif Intell*, vol. 7, p. 25, 1993.

[35] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction*. Springer, 2015.

[36] M. Torzoni, L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "SHM under varying environmental conditions: an approach based on model order reduction and deep learning," *Comput Struct*, vol. 266, p. 106790, 2022.

[37] M. Torzoni, L. Rosafalco, and A. Manzoni, "A combined model-order reduction and deep learning approach for structural health monitoring under varying operational and environmental conditions," *Eng Proc*, vol. 2, no. 1, p. 94, 2020.

[38] A. Pandey and M. Biswas, "Damage detection in structures using changes in flexibility," *J Sound Vib*, vol. 169, no. 1, pp. 3–17, 1994.

[39] A. Teughels, J. Maeck, and G. De Roeck, "Damage assessment by fe model updating using damage functions," *Comput Struct*, vol. 80, no. 25, pp. 1869–1879, 2002.

[40] M. G. Kapteyn, J. V. R. Pretorius, and K. E. Willcox, "A probabilistic graphical model foundation for enabling predictive digital twins at scale," *Nat Comput Sci*, vol. 1, no. 5, pp. 337–347, 2021.

[41] J. Hesthaven and S. Ubbiali, "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *J Comput Phys*, vol. 363, pp. 55–78, 2018.

[42] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighborhood structure," *J Mach Learn Res*, vol. 2, pp. 412–419, 2007.

[43] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*, pp. 539–546, 2005.

[44] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," in *Predicting structured data*, p. 10, MIT Press, 2006.

[45] P. Battaglia, J. B. C. Hamrick, V. Bapst, A. Sanchez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. E. Dahl, A. Vaswani, K. Allen, C. Nash, V. J. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu, "Relational inductive biases, deep learning, and graph networks," 2018. `https://arxiv.org/pdf/1806.01261`.

[46] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J Mach Learn Res*, vol. 9, pp. 249–256, 2010.

[47] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Int Conf Learn Represent*, vol. 3, pp. 1–13, 2015.

[48] F. Chollet *et al.*, "Keras," 2015. `https://keras.io`.

[49] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *Int Stat Rev*, vol. 57, no. 3, pp. 238–247, 1989.

[50] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Am Stat*, vol. 46, no. 3, pp. 175–185, 1992.

[51] C.-L. Liu and Q.-H. Chen, "Metric-based semi-supervised regression," *IEEE Access*, vol. 8, pp. 30001–30011, 2020.

[52] J.-A. Goulet, *Probabilistic Machine Learning for Civil Engineers*. MIT Press, 2020.

[53] A. Corigliano and S. Mariani, "Parameter identification in explicit structural dynamics: Performance of the extended kalman filter," *Comput Methods Appl Mech Eng*, vol. 193, no. 36-38, pp. 3807–3835, 2004.

[54] S. Eftekhar Azam, S. Mariani, and N. Attari, "Online damage detection via a synergy of proper orthogonal decomposition and recursive bayesian filters," *Nonlinear Dyn*, vol. 89, no. 2, pp. 1489–1511, 2017.

[55] N. M. Newmark, "A method of computation for structural dynamics," *J Eng Mech Divis*, vol. 85, no. 3, pp. 67–94, 1959.

[56] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "Combined model order reduction techniques and artificial neural network for data assimilation and damage detection in structures," in *Computational Sciences and Artificial Intelligence in Industry: New Digital Technologies for Solving Future Societal and Economical Challenges*, pp. 247–259, Springer International Publishing, 2022.

[57] G. Kerschen and J. C. Golinval, "Physical interpretation of the proper orthogonal modes using the singular value decomposition," *J Sound Vib*, vol. 249, no. 5, pp. 849–865, 2002.

[58] G. Kerschen, J. C. Golinval, A. Vakakis, and L. A. Bergman, "The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview," *Nonlinear Dyn*, vol. 41, no. 1, pp. 147–169, 2005.

[59] G. Gobat, A. Opreni, S. Fresca, A. Manzoni, and A. Frangi, "Reduced order modeling of nonlinear microstructures through proper orthogonal decomposition," *Mech Syst Signal Process*, vol. 171, p. 108864, 2022.

[60] J. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer, 2016.

[61] F. Negri, "redbkit, version 2.2," 2016. `http://redbkit.github.io/redbKIT`.

[62] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, Springer New York Inc., 2001.

[63] F. Chinesta, E. Cueto, E. Abisset-Chavanne, J. Duval, and F. Khaldi, "Virtual, digital and hybrid twins: A new paradigm in data-based engineering and engineered data," *Arch Comput Methods Eng*, vol. 27, pp. 105–134, 2020.

[64] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J Mach Learn Res*, vol. 12, pp. 2825–2830, 2011.

[65] T. Arvidsson and J. Li, *Dynamic analysis of a portal frame railway bridge using frequency dependent soil structure interaction.* Master thesis, KTH Royal Institute of Technology, 2011.

[66] M. Ülker-Kaustell, *Some aspects of the dynamic soil-structure interaction of a portal frame railway bridge.* PhD thesis, KTH Royal Institute of Technology, 2009.

[67] European Committee for Standardization, "Part 2: Traffic loads on bridges," in *EN 1991-2 Eurocode 1: Actions on structures*, pp. 66–74, 2003.

Appendix A. Details on the convolutional units and dense layers

For completeness, in this Appendix the algorithmic details of the convolutional units and dense layers used in this work are reported.

During training, the learning algorithm tunes the NN parameters in $\mathbf{\Omega}$, by iteratively minimizing the loss function over the I_P pairs. At each iteration, the NN parameters are updated after having processed a mini-batch made of B instances. The tunable parameters $\mathbf{\Omega}$, here include the convolution kernel filters $\mathbf{\Theta}$ and the weights $\boldsymbol{\omega}$ of the dense layers.

Each convolutional unit consists of: a convolutional layer $\mathcal{C}$, a nonlinear activation function $\mathcal{N}$, a batch normalization $\mathcal{B}$ and a 1D max pooling layer $\mathcal{P}$. What described in the following applies to all three convolutional units.

Let $\mathbf{U}^{k-1} = \{\mathbf{U}_b^{k-1}\}_{b=1}^{B} = \{(\mathbf{u}_{1b}^{k-1}, \ldots, \mathbf{u}_{N_k b}^{k-1})\}_{b=1}^{B} \in \mathbb{R}^{L_{k-1} \times N_{k-1} \times B}$ be the input tensor of the k-th convolutional layer $\mathcal{C}_k$, with $k = 1, 2, 3$. The relevant output of $\mathcal{C}_k$ is given by:

$$\boldsymbol{\varkappa}_{nb}^{k} = \sum_{m=1}^{N_{k-1}} \mathbf{\Theta}_n^{km} * \mathbf{u}_{mb}^{k-1} \ , \quad n = 1, \ldots, N_k \ , \quad b = 1, \ldots, B \ , \tag{A.1}$$

and assembled as $\boldsymbol{\mathcal{Z}}^k = \{(\boldsymbol{\varkappa}_{1b}^k, \ldots, \boldsymbol{\varkappa}_{N_k b}^k)\}_{b=1}^{B} \in \mathbb{R}^{L_k^* \times N_k \times B}$. In Eq. (A.1): $* : (\mathbb{R}^{H_k} \times \mathbb{R}^{L_{k-1}}) \to \mathbb{R}^{L_k^*}$ is the convolution operation of kernel size H_k; $\mathbf{\Theta}_n^k = \{\mathbf{\Theta}_n^{km}\}_{m=1}^{N_{k-1}} \in \mathbb{R}^{H_k \times N_{k-1}}$ are the kernel filters, that the n-th channel simultaneously applies to the input tensor of the b-th instance within the current mini-batch; $\mathbf{\Theta}^k = \{\mathbf{\Theta}_n^k\}_{n=1}^{N_k} \in \mathbb{R}^{H_k \times N_{k-1} \times N_k}$ is the overall filters set. Then, $\mathbf{\Theta} = \{\mathbf{\Theta}^1, \mathbf{\Theta}^2, \mathbf{\Theta}^3\}$ collects all the tunable parameters of $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$.

The linear activations produced by the repeated convolutions are then run through a nonlinear activation function, which in this case is the hyperbolic tangent function, to provide:

$$\bar{\varkappa}_{nb}^{kl} = \mathcal{N}(\varkappa_{nb}^{kl}) = \tanh(\varkappa_{nb}^{kl}) \ , \quad l = 1, \ldots, L_k^* \ . \tag{A.2}$$

Batch normalization is used to mitigate the gradient issues typically encountered when training deep NN architectures, and thus stabilize the training dynamics by re-centering and re-scaling the outputs of the last hidden layer across the relevant mini-batch, according to:

$$\hat{\varkappa}_{nb}^{kl} = \mathcal{B}_{kn}(\bar{\varkappa}_{nb}^{kl}) = \gamma_{\mathcal{B}_{kn}} \cdot \frac{\bar{\varkappa}_{nb}^{kl} - \mu_{\mathcal{B}_{kn}}}{\sqrt{\sigma_{\mathcal{B}_{kn}}^2 + \epsilon_{\mathcal{B}}}} + \beta_{\mathcal{B}_{kn}} \ , \quad l = 1, \ldots, L_k^* \ , \tag{A.3}$$

where: $\mu_{\mathcal{B}_{kn}}$ and $\sigma_{\mathcal{B}_{kn}}$ are the mean and the variance computed across the batch of the activation values relevant to the n-th feature, which are used to enforce the inputs to comply with a unit normal distribution; during inference, $\mu_{\mathcal{B}_{kn}}$ and $\sigma_{\mathcal{B}_{kn}}$ are computed as a moving average of, respectively, the mean and standard deviation of the batches seen during training, to give more importance to the last iterations; $\epsilon_{\mathcal{B}}$ is a small constant useful to avoid numerical instabilities; $\gamma_{\mathcal{B}_{kn}}$ and $\beta_{\mathcal{B}_{kn}}$ are a scaling factor and an offset factor, respectively, that are learned during training in order to restore the flexibility of the NN.

Finally, the pooling layer reduces the dimensionality of $\hat{\boldsymbol{\mathcal{Z}}}^k = \{(\hat{\boldsymbol{\varkappa}}_{1b}^k, \ldots, \hat{\boldsymbol{\varkappa}}_{N_k b}^k)\}_{b=1}^{B} \in \mathbb{R}^{L_k^* \times N_k \times B}$ by down sampling the detection of nearby features [10]. Here, the 1D max pooling is used to obtain $\mathbf{U}^k \in \mathbb{R}^{L_k \times N_k \times B}$ from $\hat{\boldsymbol{\mathcal{Z}}}^k$, by choosing a stride equal to 2, in order to subsample within a 1D neighborhood in time and resulting in an output size equal to $L_k = L_k^*/2$.

In a dense layer $\mathcal{D}_k$, each neuron is fully-connected to its input layer, and it provides a scalar output ruled by the composition of a nonlinear activation function with a linear mapping. Denoting by $\mathbf{A}^{k-1} = \{\mathbf{A}_b^{k-1}\}_{b=1}^{B} = \{(a_{1b}^{k-1}, \dots, a_{N_{k-1}b}^{k-1})\}_{b=1}^{B} \in \mathbb{R}^{N_{k-1} \times B}$ the input mini-batch for the k-th dense layer, the output is computed as:

$$a_{nb}^{k} = \mathcal{N}(\boldsymbol{\omega}_n^k \cdot \mathbf{A}_b^{k-1} + \beta_n^k) \ , \quad n = 1, \dots, N_k \ , \quad b = 1, \dots, B \ , \tag{A.4}$$

and assembled as $\mathbf{A}^k = \{(a_{1b}^k, \dots, a_{N_k b}^k)\}_{b=1}^{B} \in \mathbb{R}^{N_k \times B}$. In Eq. (A.4): $\boldsymbol{\omega}_n^k \in \mathbb{R}^{N_{k-1}}$ are the connection weights ruling the linear mapping; β_n^k is the bias term; $\boldsymbol{\omega}^k = \{(\boldsymbol{\omega}_n^k, \beta_n^k)\}_{n=1}^{N_k} \in \mathbb{R}^{(N_{k-1}+1) \times N_k}$ is the overall filters set of $\mathcal{D}_k$; $\mathcal{N}$ is the nonlinear activation function, which is chosen as the hyperbolic tangent function. The set $\boldsymbol{\omega} = \{\boldsymbol{\omega}^4, \boldsymbol{\omega}^5, \boldsymbol{\omega}^6\}$ collects all the tunable parameters of $\mathcal{D}_4$, $\mathcal{D}_5$ and $\mathcal{D}_6$.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

19/2024 Torzoni, M.; Manzoni, A.; Mariani, S.
A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks

17/2024 Fois, M.; de Falco, C.; Formaggia, L.
A semi-conservative depth-averaged Material Point Method for fast flow-like landslides and mudflows

16/2024 Domanin D. A.; Pegoraro M.; Trimarchi S.; Domanin M.; Secchi P.
Persistence diagrams for exploring the shape variability of abdominal aortic aneurysms

15/2024 Vaccaro, F.; Mauri, A.G.; Perotto, S.; Brivio, S.; Spiga, S.
Modeling and simulation of electrochemical and surface diffusion effects in filamentary cation-based resistive memory devices

14/2024 Zappon, E.; Salvador, M.; Piersanti, R.; Regazzoni, F.; Dede', L.; Quarteroni, A.
An integrated heart-torso electromechanical model for the simulation of electrophysiological outputs accounting for myocardial deformation

12/2024 Zingaro, A.; Ahmad, Z.; Kholmovski, E.; Sakata, K.; Dede', L.; Morris, A.K.; Quarteroni, A.; Trayanova, N.A.
A comprehensive stroke risk assessment by combining atrial computational fluid dynamics simulations and functional patient data

Antonietti, P.F.; Corti, M.
Numerical modelling of protein misfolding in neurodegenerative diseases a computational study

11/2024 Antonietti, P.F.; Corti, M.
Numerical modelling of protein misfolding in neurodegenerative diseases: a computational study

09/2024 Leimer Saglio, C. B.: Pagani, S.; Corti, M.; Antonietti, P. F.
A high-order discontinuous Galerkin method for the numerical modeling of epileptic seizures

10/2024 Capuano E.; Regazzoni F.; Maines M.; Fornara S.; Locatelli V.; Catanzariti D.; Stella S.; Nobile F.; Del Greco M.; Vergara C.
Personalized Computational Electro-mechanics Simulations to Optimize Cardiac Resynchronization Therapy