ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Full Strain Matrix Estimation in Thin-Walled Structures With Recurrent Inpainting Model

Ángel Cruz-Alonso
Instituto Nacional de Técnica Aeroespacial (Spain),
Madrid 28048, Spain;
Universidad Rey Juan Carlos (Spain),
Madrid 28933, Spain
e-mail: acrualo@inta.es

Félix Terroba
Instituto Nacional de Técnica Aeroespacial (Spain), Madrid 28048, Spain
e-mail: terrobarf@inta.es

Alfredo Cuesta-Infante[1]
Universidad Rey Juan Carlos (Spain),
Madrid 28933, Spain
e-mail: alfredo.cuesta@urjc.es

Thin-walled structures are ubiquitous in industries such as automotive, civil engineering, consumer electronics, or medical devices; and many times these structures, or a significant part of them, can be approximated by a plate as in aerospace and shipbuilding. In order to prevent damages and increase safety, "on-condition" maintenance is increasingly being used due to the nowadays ability to continuously sensing and processing data in real time. A key feature to assess the probability of damage is the strain caused by loads. In this article, our goal is to estimate the strain in the whole structure based on measurements that only capture a 1.2% of its total surface. We show that the problem is equivalent to reconstructing an image with 98.8% missing pixels and present a novel procedure referred to as the recurrent inpainting model *(RIM). We use finite element methods to simulate a thin-walled structure under different loads and create a large data set of instances. Then, we use RIM to carry out the reconstruction task along with tests of robustness against sensor failure, transferability to other sensor morphologies, and generalization to 3D hollow structures. The results in all the tasks clearly outrank the next best deep learning architecture.* [DOI: 10.1115/1.4071388]

Keywords: artificial intelligence, data-driven engineering, machine learning for engineering applications, physics-based simulations

1 Introduction

Currently, many of the most advanced structures from the point of view of structural efficiency used in sectors such as aeronautics, space, or high-performance shipbuilding share the following characteristics: (1) the use of high-performance composite materials, (2) the use of thin-walled structures from 0.5 up to 10 mm thick, and (3) the use of flat plates or membranes with a radius of curvature in the order of 10^2 m in approximately the 90% of the surfaces.

Examples can be found in the structure of an aircraft wing [1] or in the hull of a high-performance ship [2], in which most of the elements are either flat with a large radius of curvature together with a very small thicknesses. Moreover, such a "semi-monocoque" structure is currently the most commonly used in sectors where the aim is to obtain elements that are optimized from a structural and mass point of view [3–5].

In addition, it is important to mention that these types of structures are currently designed with the "damage tolerant" philosophy, i.e., the structure must be able to resist the occurrence of damage and its evolution without compromising its integrity before the damage is detected and repaired. Until recently, the detection of damage and its subsequent repair was linked to scheduled or calendar maintenance consisting of periodic inspections. In order to save costs while ensuring safety, "on-condition" maintenance is nowadays the preferred option, which is based on the continuous monitoring of the structure and the detection and/or estimation of the presence of damage in real time [6].

To this end, the first stage is to instrument the structure of interest, that is, to install, embed, or attach permanent sensors or monitoring devices in a distributed pattern to provide measurements of vibrations, deformations, and other characteristics at least in near real time. The collected data are then analyzed to diagnose any possible damage or failure in time [7].

Instrumentation challenges can vary depending on the subject; an outdoor infrastructure will suffer from environmental factors and needs power supply, while a composite panel may require careful integration of sensors to preserve its structural integrity [8]. In any case, the distribution of sensors, in terms of number and pattern, referred to as *morphology*, is a key issue. The choice must ensure complete coverage of the entire structure, which means that the data collected are sufficient for further analysis. Such coverage can be compromised by two factors: (1) sensor malfunction and (2) data corruption. A solution is to infer the missing values from the collected data [9,10].

Usually, finite element methods (FEMs) [11] have been widely employed to model structural behavior under various conditions and predict responses with high accuracy. This approach enables researchers to interpolate across diverse conditions and scales [12,13]. Although FEM-based simulations may be time consuming, as in highly nonlinear systems or complex geometries, they provide a large synthetic but very realistic dataset that can be used as a ground truth for training deep learning (DL) models, which are particularly well suited for interpolation. The advantages

[1]Corresponding author.

Manuscript received August 1, 2025; final manuscript received January 15, 2026; published online April 10, 2026. Assoc. Editor: Johann Guilleminot.

of this approach are as follows: (1) ground truth masking simulates the data collected only by sensors; over these, (2) ground truth poisoning simulates the data corruption such as missing or noisy measurements; and (3) DL model inference is several orders of magnitude faster than FEM-based simulations [10].

In this article, we follow this approach. Our goal is to estimate the value of a given feature at any point of the geometry from a reduced set of measurements. Specifically, the geometry is a thin-walled plate and the feature considered in the strain suffered when different loads are applied on it. Then, we cast the problem as one of *extreme* inpainting, in which around 98.8% of the image is missing and address the following challenges: (1) data representation, for no-strain is different from no-data, (2) accuracy of the reconstruction, (3) robustness to sensor failure, (4) transferability of the model to other morphologies without retraining nor fine-tuning, and (5) extension to 3D hollow structures. To successfully accomplish all of them, we introduce the *recurrent inpainting model* (RIM), which encompasses the following contributions:

(1) A preprocessing protocol to turn a strain matrix into an image.
(2) A novel deep learning architecture that leverages the autoencoder (AE) with skip connections, commonly known as U-net, and incorporates a recurrent mechanism derived from the denoising diffusion probabilistic models (DDPMs).
(3) Two new loss functions that are added to the usual mean squared error (MSE) to provide a better learning signal.
(4) Extensive experimentation that includes: (1) a comparative of the U-net architecture with other generative ones, specifically β-variational autoencoder (VAE), conditional generative adversarial networks (CGANs), and diffusion models; (2) a comparative of U-net architecture with RIM to assess the performance in each one of the challenges listed earlier. (3) An ablation study on the components in RIM added to the U-net.
(5) A thorough analysis and discussion of the models evaluated and the results attained.

RIM outperforms these approaches in all the challenges posed. Additionally, the dataset created and used in this study, along with the code of RIM, is available online.[2]

The rest of the article is organized as follows. In Sec. 2, we describe the problem and address the data representation challenge. The model proposed, how it makes the inference, and how it is trained are detailed in Sec. 3. Section 4 covers the remaining challenges, namely, reconstruction of the strain matrix, robustness of RIM to sensor failure, its transferability to other morphologies, and extension to 3D hollow structures. The related work is discussed in Sec. 5. Finally, Sec. 6 points out the conclusions of this study.

2 Problem Statement

When instrumenting a structure, the number of sensors that can be installed is a frequent limitation that has a twofold impact on its coverage: (1) directly, for there will only be actual measurements where they are placed, and (2) indirectly, since everywhere else has to be interpolated. The accuracy of the interpolation will depend on the number of sensors and their arrangement, commonly referred to as *morphology* in the on-condition maintenance context.

In this article, we focus on thin-walled structures and show that it is possible to get an accurate interpolation leveraging state-of-the-art deep learning models and FEM-based simulations.

2.1 Test Bench. We consider a square plate of 50 cm × 50 cm, instrumented with Fiber Bragg Gratings (FBGs). The geometry of the structure is modeled in Patran[3], with 32×32 elements specifying the following characteristics: Young's modulus E (resistance to deformation under stress), shear modulus G (response to shear stress), and Poisson's ratio v (ratio of lateral strain to axial strain when the structure is stretched). In particular, we set

$$E = 70\,\text{GPa}, \quad G = 26\,\text{GPa}, \quad v = 1/3$$

which corresponds to the standard aluminum. The FBG provides strain measurements in the X- and Y-axis as follows: a grating inscribed within the core of an optical fiber reflects one particular wavelength of light and transmits the rest; whenever a strain is applied on the fiber, the spacing of the grating changes the reflected wavelength proportionally.

Loads are applied to this structure in different positions and of different magnitude given in Newtons (N) to create variety of deformation patterns. Specifically, the number of loads is sampled from a uniform distribution between 1 and 6. Each one of them is placed in a position (x, y) sampled from a bivariate uniform distribution within the bounds of the plate. The magnitude is sampled from a continuous uniform distribution between 0 and 100. Additionally, physical constraints are imposed to the ends of the structure as boundary conditions, mimicking the way these structures are typically anchored. Naming the four sides of the plate clockwise as A, B, C, and D, the constraint applied is sampled from a uniform distribution over all the possibilities {None, A, B, C, D, AB, . . . , ABCD}, where each letter represents an anchor on the corresponding side. We remark that this test bench is not physically implemented. The choice of FBG is motivated by their extreme sensitivity, capable of detecting microstrains, which matches the high fidelity provided by FEM simulations. Moreover, it would also be a current option in industry. The finite element analysis is carried out by MSC-Nastran.[4] The results of this simulation are two strain matrices $\epsilon_{xx}, \epsilon_{yy} \in \mathbb{R}^{32\times32}$, which serve as *raw* ground truth. We generate a total of 8000 strain matrices equally distributed between both types.

A sketch of the test bench and its morphology is shown in Fig. 1(*a*). The sensors oriented in the X-axis provide the strain matrix corresponds to ϵ_{xx}. Similarly, those oriented in the Y-axis provide ϵ_{yy}.

2.2 Data Representation. Our approach consists of eliminating all the values in a strain matrix where there are no sensors and learning to reconstruct them from the sensor readings. This procedure raises the problem of representing "nonexistent data," which is different from "zero strain," since the latter could be a sensor reading. For this reason, the strain matrices obtained are not useful in their current form.

Our solution is to encode the strain values into a normalized (R, G,B) tuple according to a color map. To achieve this, every *raw* strain matrix undergoes the following transformation pipeline.

(1) Normalization to the unit interval. This step learns the maximum and minimum strain values in every matrix and scales linearly all its elements, resulting a matrix in $[0, 1]^{32\times32}$. We remark that this is different from normalizing both training, validation and test splits with respect to the maximum or minimum strain values attained in the whole data set of *raw* matrices.
(2) Encoding to the Viridis color map. Viridis ranges from deep purple color, with RGB = (68, 1, 84), to bright yellow, RGB = (253, 231, 37). This color map is color-blindness robust and perceptually uniform. The choice of Viridis is arbitrary, and any other color map would do as long as it complies with the same properties. The outcome of this step is an array of three matrices: matrix $R \in [68, 253]^{32\times32}$, matrix $G \in [1, 231]^{32\times32}$, and matrix $B \in [84, 37]^{32\times32}$.

[2] https://github.com/ancral/rim.
[3] https://hexagon.com/products/patran.
[4] https://hexagon.com/es/products/product-groups/computer-aided-engineering-software/msc-nastran.

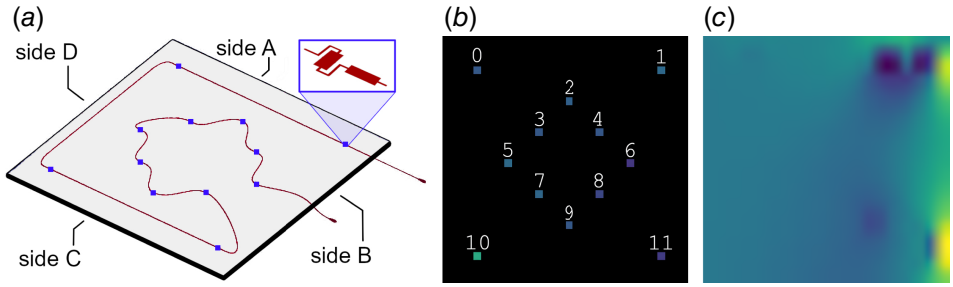

Fig. 1 (*a*) An arrangement of 12 FBG sensors. (*b*) Readings of strain are only taken in those elements with sensors in them. Sensors are numbered with an id. (c) Example of strain matrix ϵ resulting from simulation of FEM.

(3) Divide by 255. Hence, matrix $R \in [0.267, 0.992]^{32\times32}$, matrix $G \in [0.004, 0.906]^{32\times32}$ and matrix $B \in [0.329, 0.145]^{32\times32}$.

The resulting R, G, and B matrices are used as ground truth. Additionally, to obtain a representation of the FBG readings due to the loads applied to the structure in a particular instance, we set every pixel to pitch black, i.e., [0, 0, 0] except for those pixels that correspond to a sensor, where we keep the ground truth value. We show an example of sensor reading (the input to RIM) and its corresponding ground truth (the intended target) in Figs. 1(*b*) and 1(*c*), respectively. Particularly, in these figures, it can be noticed that a *missing* pixel is completely different from another with strain = 0. The former is any of the pitch-black pixels in Fig. 1(*b*), whereas the latter corresponds to any pixel with a green-blue tone in Fig. 1(*c*). Hence, the RGB codification is a valid solution to capture both, whereas using the scalar values from the readings is not.

3 Model Description

The data representation turns the interpolation (also referred to as reconstruction) problem of finding the strain in all elements of the geometry into an *extreme* inpainting problem, in which 1012 out of 1024 of the pixels (about 98.8%) are missing. The state-of-the-art neural architectures for this task are AEs, CGANs, and DDPMs. In this section, we first check their performance to assess the necessity of a new one (RIM) and then introduce the inference and training.

For the rest of the article, we use the following notation. Symbol $\mathbb{1}$ represents the indicator function, which returns 1 if its argument is true and 0 otherwise. We use lowercase letters x to refer to the pixels (even if these have three channels) of an image $\mathbf{x}$, denoted in lowercase bold letters. We indicate with the *hat* ($\hat{\cdot}$) any scalar or multidimensional vector estimated by a model. Let $\mathbf{x}_T$ be the input image, i.e., the 1 with the 12 FBG readings encoded in 0-to-1 RGB and [0, 0, 0] everywhere else. Let $\mathbf{x}_0$ be its target image, let $\hat{\mathbf{x}}_0$ be the image estimated by a model for the input $\mathbf{x}_T$, and let $\hat{\mathbf{x}}_t$ be the image estimated at any step $0 \leq t < T$ if the model's mechanism requires it.

3.1 Discussion About State-of-the-Art Methods

3.1.1 Autoencoders. There are two prominent architectures, namely, U-net and VAEs. The U-net key feature is the skip connections that forward visual features from the encoder layers directly to the corresponding decoder layers [14]. VAEs are generative machines that first encode the input into a multivariate normal (MVN) distribution from which the latent representation is sampled. The decoder learned is a deterministic transformation of the MVN into the distribution of the target images, hence the sampled latent representation turns into a sample of the target. A convenient modification of VAE is the β-VAE, in which hyperparameter β controls the trade-off between reconstruction accuracy and the disentanglement of the learned latent representations. The former means that two elements of the latent representation have an effect on different parts of the output [15].

3.1.2 CGAN. It is also a generative model that starts from a random noise vector sampled from a multivariate uniform to which another vector with fixed values is appended. The result is introduced to a decoder architecture whose learning is done with a discriminator and the pretext task of distinguishing generated images from real ones. Since the input vector is not fully random, the image generated is conditioned to the fixed part appended [16].

3.1.3 DDPM. This is the most recent approach regarding image generation with outstanding results in terms of realistic looking. It consists of a sequence of T steps that gradually transform an image into Gaussian white noise. At every step t, each pixel of the image produced x_t is sampled from the conditional distribution $q(X_t|X_{t-1} = x_{t-1})$, where q is chosen to be a normal with mean $\mu_t = (\sqrt{1-\beta_t})x_{t-1}$ and variance $\sigma_t^2 = \beta_t$. According to the so-called "parametrization trick," a sampled pixel $\tilde{x}_t$ can be expressed as follows:

$$\tilde{x}_t = \left(\sqrt{1-\beta_t}\right)x_{t-1} + \left(\sqrt{\beta_t}\right)\varepsilon \tag{1}$$

where ε is sampled from a standard normal distribution. Hyperparameter β_t ranges from 0 to 1 according to a schedule that depends on the number of steps. Due to the parametrization trick, the only random element at each step is ε, and DDPM trains a neural network model to estimate ε. The inference starts from Gaussian white noise and the neural network estimates *what* has to be removed. Thus, the image is progressively cleaner in a backwards process [17].

3.1.4 Comparison. To the best of our knowledge, there is no previous work with such a rate of missing pixels. Therefore, the first approach should be checking the performance of the models cited earlier when reconstructing a strain matrix from the readings of 12 FBG. To this end, we trained and fine-tuned a U-net, a β-VAE, and a CGAN using the MSE between the estimated image and the ground truth as loss function. The results are shown in Table 1.

The test clearly selects U-net as the most promising model. A plausible reason is that skip connections send the local information

Table 1 Comparison of state-of-the-art models for *inpainting* strain matrices

	U-net	β-VAE	CGAN	DDPM
MSE	0.002	0.016	0.035	>1

of the FBG readings to the decoder and the layers of the latter learn to complete the image. Yet, it is questionable whether these models are well-tuned for the task. The hyperparameter space of neural network models is immense. The most common ones, in the sense of being task-independent, are those due to the network architecture (number of layers, number of neurons in each layer, type of activation function in each layer, etc.) and to the optimizer (momentum, learning rate, weight decay, etc.). In addition, each model has its own hyperparameters, for example, the value of β in β-VAEs or the scheduling in DDPM. In our experiments, we conducted an extensive battery of tests, from which we obtained the settings that were finally used to produce the Table 1. Besides, these results lead us to offer an plausible explanation based on the specific features of each model. The architecture of β-VAE lacks skip connections and the encoder faces the task of encoding 1012 black plus 12 nonblack pixels into a mean and a variance vectors. Thus, it is likely that both will shrink to zero, which makes learning the decoder more difficult than the U-net's, yielding poorer performance. CGAN conditions the random input to the input image. One could hypothesize that the encoder may *ignore* the randomness and reconstruct only from the conditioning image. Even if so, the discriminator only estimates the likelihood of the generated image being a true one, which to the light of the results is not enough information to obtain an accurate reconstruction. Finally, during the training of the DDPM, the ground truth images are progressively turned into Gaussian white noise. Then, in inference, we generate new Gaussian white noise, restore the known pixels to condition the output, and produce the estimated image. The performance is more than 30 times worse than CGAN, and more than 500 times worse than U-net, suggesting that the white noise is poisoning the no-data representation.

3.2 Derivation of a New Model. We aim at improving the U-net outcome leveraging its key feature, the skip connections, and overcoming the issues that the rest of models present. Keeping the U-net architecture excludes the β-VAE, but it could be the generator of a CGAN. However, as claimed earlier, the discriminator does not provide the best learning signal. The DDPM already includes the U-net, so we propose a different mechanism that is derived from Eq. (1). First, notice that Eq. (1) produces a normally distributed sample $\tilde{x}_t$ only if ε comes from the standard normal distribution. Otherwise, Eq. (1) is just a parametrization of $\tilde{x}_t$ in terms of x_{t-1}, β_t, and ε. Solving for x_{t-1}, we obtain

$$x_{t-1} = \left(\frac{1}{\sqrt{1-\beta_t}}\right)\left(\tilde{x}_t - \sqrt{\beta_t}\cdot\varepsilon\right) \tag{2}$$

Then, we approximate ε using the U-net, represented as the parametric function $f_{\boldsymbol{\theta}}(\cdot)$, and obtain

$$\hat{x}_{t-1} = \left(\frac{1}{\sqrt{1-\beta_t}}\right)\left(\tilde{x}_t - \sqrt{\beta_t}\cdot f_{\boldsymbol{\theta}}(\tilde{x}_t)\right) \tag{3}$$

At this point, a first option is to set $T = 1$, which is just introducing the input image and let the U-net produce an estimation of the inpainting. Setting $T > 1$ leads to a *reminiscent DDPM* in which the output at t is the input at $t-1$, since the process is backward in time. Additionally, if we consider the whole image instead of a single pixel and incorporate this recurrent mechanism, Eq. (3) is rewritten as follows:

$$\hat{\mathbf{x}}_{t-1} = \left(\frac{1}{\sqrt{1-\beta_t}}\right)\left(\hat{\mathbf{x}}_t - \sqrt{\beta_t}\cdot f_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\right) \tag{4}$$

Besides, in order to keep the FBG readings and to aid in the reconstruction, we incorporate a *restore–enhance–smooth* (RES) module, which works as follows.

3.2.1 Restore. Let us rename the output of Eq. (4) as $\mathbf{y}_t$. Then, this step overwrites those pixels of $\mathbf{y}_t$ that correspond to the locations of the FBG according to the morphology with the values in the input $\mathbf{x}_T$. The rest of the pixels are clipped to [0, 1]. Formally, this step is carried out for each pixel by the *restoration* function R, defined as follows:

$$R(y_t, x_T) = \begin{cases} x_T & \text{if} \quad x_T \neq [0,0,0], \\ \text{Clip}(y_t) & \text{otherwise} \end{cases} \tag{5}$$

Let $z_t = R(y_t, x_T)$, then restoration in all the pixels of $\mathbf{y}_t$ produces the image $\mathbf{z}_t$.

3.2.2 Enhance. This step aims at emphasizing those pixels in the upper and lower quintiles of luminance ℓ, which is computed for each pixel of $\mathbf{z}_t$ as follows:

$$\ell = 0.2126\cdot z_{\text{R}} + 0.7152\cdot z_{\text{G}} + 0.0722\cdot z_{\text{B}} \tag{6}$$

where $z = [z_{\text{R}}, z_{\text{G}}, z_{\text{B}}]$ represents the RGB channels of a pixel x. Thus, for the Viridis color map, we get the upper and lower quintiles for every RGB channel as follows:

$$\begin{cases} \ell_{Q5} = \ell \cdot \mathbb{1}(\ell > 0.8)\cdot[0.992, 0.906, 0.145] \\ \ell_{Q1} = \ell \cdot \mathbb{1}(\ell > 0.2)\cdot[0.267, 0.004, 0.329] \end{cases}$$

The *enhancement* function E adds z's difference between ℓ_{Q5} and ℓ_{Q1} to z but with two caveats: (1) it is only applied from time-step $t = \tau$ down to $t = 1$, and (2) such a difference is multiplied by an increasing hyperparameter that depends on the time-step. The result is also clipped to [0, 1].

$$E(z_t, t) = \text{Clip}\big(z_t + \mathbb{1}(t \leq \tau)\cdot(\tau - t)\cdot\gamma\cdot\big(\ell_{Q5} - \ell_{Q1}\big)\big) \tag{7}$$

Let $w_t = E(z_t, t)$, then enhancement in all the pixels of $\mathbf{z}_t$ produces the image $\mathbf{w}_t$.

3.2.3 Smooth. Finally, we apply a Gaussian smoothing to the image $\mathbf{w}_t$ using the convolution with a 7×7 Gaussian kernel $\mathbf{G}$, with mean zero and standard deviation $\sigma = 2$. The *smoothing* function S adds the difference between such a convolution and the enhanced image $\mathbf{w}_t$ multiplied by the hyperparameter ζ. Mathematically, it is expressed as follows:

$$S(\mathbf{w}_t) = \mathbf{w}_t + \zeta(\mathbf{w}_t \star \mathbf{G} - \mathbf{w}_t) \tag{8}$$

The outcome of the smoothing function is the estimated image in time-step t, which is also the input for the time-step $t-1$, i.e., $\hat{\mathbf{x}}_{t-1} = S(\mathbf{w}_t)$.

3.2.4 Remarks. RIM has two distinctive features with respect to U-net (the closest competitor) and DDPM (the foundation of the method proposed).

First, the RES module is time dependent, which makes it a key feature of the proposed method, as opposed to the alternative of a U-net with postprocessing. The option of a U-net that is used several times, feeding it in iteration t with the output of the previous iteration, does not make sense for two reasons: (1) the single forward pass in the U-net already produces good results and (2) as the U-net parameters are frozen in inference, its response in the second and subsequent iterations is not guaranteed since its input in them is completely different. On the contrary, our proposal interleaves the output of the U-net with R, E, and S at each iteration.

Second, the output of the U-net is transformed via Eq. (4), which is derived from the parametrization of sampling from a normal distribution as in DDPM in Eq. (1). But DDPM is intrinsically different to RIM because (1) the ground truth in DDPM is the noise added every time, while the ground truth in RIM is the target image and (2) DDPM is trained in a forward and back-propagation schema, whereas RIM's is recursive-forward T times and back-propagation-through-time (see Sec. 3.4, for a detailed description of the training process).

3.3 Inference. Algorithm 1 shows the sequence of steps due to a single forward pass, which comprises iterating recurrently T times over the inpaint–restore–enhance–smooth steps.

Algorithm 1 Inference With Recurrent Inpainting Model

Require: Input image $\mathbf{x}_T$
1: $\hat{\mathbf{x}}_T \leftarrow \mathbf{x}_T$
2: **for** $t : T$ down to 1 **do**
3: Compute $\mathbf{y}_t$, such that every pixel $y_t = \left(\sqrt{1-\beta_t}\right)^{-1}\left(\hat{\mathbf{x}}_t - \sqrt{\beta_t}\cdot f_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_t)\right)$
4: Compute $\mathbf{z}_t$, such that every pixel $z_t = R(y_t, x_T)$
5: Compute $\mathbf{w}_t$, such that every pixel $w_t = E(z_t, t)$
6: Compute $\hat{\mathbf{x}}_{t-1} = S(\mathbf{w}_t)$
7: **end for**
8: **return** $\hat{\mathbf{x}}_0$

3.4 Training. The only trainable parameters are those in the U-net, i.e., θ in $f_{\boldsymbol{\theta}}(\cdot)$. Since the rest of functions involved are differentiable, it is possible to apply gradient descent after a complete (in the sense of running the T iterations) forward pass. In order to assess the performance of the model with respect to those in Table 1, the MSE loss is used. Additionally, we propose two new terms for the loss: one that focuses on the accuracy in the lower and upper sides (the *extremes*) of the color map and another that focuses on the similarity between target and estimation directional derivatives.

3.4.1 Extremes Loss. Let $[x_R, x_G, x_B]$ be the RGB channels of a pixel x from the target image $\mathbf{x}_0$, and let m be its corresponding *mask* in matrix M defined element-wise as follows:

$$m = \mathbb{1}\left(m_L \text{ or } m_U\right), \text{ where} \tag{9}$$

$$m_L = \mathbb{1}\left((x_R \le \nu) \text{ and } (x_G \le \nu) \text{ and } (x_B \ge 1-\nu)\right), \text{and} \tag{10}$$

$$m_U = \mathbb{1}\left((x_R \ge 1-\nu) \text{ and } (x_G \ge 1-\nu) \text{ and } (x_B \le \nu)\right) \tag{11}$$

Here, the *lower and upper mask bounds*, m_L and m_U, respectively, are configurable with hyperparameter ν. Then, the extremes loss L_1 is

$$L_1(\mathbf{x}_0, \hat{\mathbf{x}}_0) = \frac{1}{N} \sum_{\substack{\forall x \in \mathbf{x}, \\ \forall \hat{x} \in \hat{\mathbf{x}}, \\ \forall m \in M}} \left(m \cdot (x - \hat{x})^2\right) \tag{12}$$

where $N = 1024$ is the number of pixels in the images.

3.4.2 Directional Derivatives Loss. Let $\mathbf{v}[i, j]$ be the element in row i and column j of a matrix $\mathbf{v}$. The derivative in the direction d of the element $\mathbf{v}[i, j]$ is defined as the difference between that element and the next one in the given direction. Thus, for direction x (columns) and y (rows), it is expressed as follows:

$$\Delta_x \mathbf{v}[i,j] = \begin{cases} \mathbf{v}[i, j+1] - \mathbf{v}[i,j] & \text{if } j < 32 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\Delta_y \mathbf{v}[i,j] = \begin{cases} \mathbf{v}[i+1, j] - \mathbf{v}[i,j] & \text{if } i < 32 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

By using Eqs. (13) and (14), we define the directional derivatives loss as follows:

$$L_2(\mathbf{x}_0, \hat{\mathbf{x}}_0) = \frac{1}{N^2} \sum_{i,j=1}^{\sqrt{N}} \left|\Delta_x \mathbf{x}_0[i,j] - \Delta_x \hat{\mathbf{x}}_0[i,j]\right| + \left|\Delta_y \mathbf{x}_0[i,j] - \Delta_y \hat{\mathbf{x}}_0[i,j]\right| \tag{15}$$

3.4.3 Loss Function Proposed. It is a weighted sum of the three losses introduced:

$$L(\mathbf{x}_0, \hat{\mathbf{x}}_0) = \lambda_0 \text{MSE}(\mathbf{x}_0, \hat{\mathbf{x}}_0) + \lambda_1(\mathbf{x}_0, \hat{\mathbf{x}}_0) L_1 + \lambda_2 L_2(\mathbf{x}_0, \hat{\mathbf{x}}_0) \tag{16}$$

This loss function, as well as each of its terms, are also used as a performance metric. Algorithm 2 shows the steps to carry out the training process.

Algorithm 2 Training Recurrent Inpainting Model

Require: Input image $\mathbf{x}_T$, Target image $\mathbf{x}_0$
Ensure: U-net suboptimal parameters θ^*
1: **repeat**
2: $\hat{\mathbf{x}}_0 \leftarrow \text{RIM} \quad (\mathbf{x}_T)$ ▷ Run inference
3: Take gradient descent step on

$$\nabla_\theta(\lambda_0 \text{MSE}(\mathbf{x}_0, \hat{\mathbf{x}}_0) + \lambda_1 L_1(\mathbf{x}_0, \hat{\mathbf{x}}_0) + \lambda_2 L_2(\mathbf{x}_0, \hat{\mathbf{x}}_0))$$

4: **until** converged

3.5 Hyperparameter Tuning. The choice of Viridis color map in the data representation stage conditions some of the expressions derived earlier. Same can be said of the specifications of the test bench, which induces the number of elements and the number of pixels in the images. For the sake of clarity, we do not introduce symbols for those, which produces some constant values. Different color maps and geometries would require minimal straightforward changes in the expressions.

Equation (4) depends on a linear scheduling for β_t, given by $\beta_t = 0.0375 \cdot t + 0.0125$, which gives $\beta_T = 0.2$ and $\beta_1 = 0.05$. The rest of the model's hyperparameters is shown in Table 2.

The gradient descent was carried out with Adam optimizer, 100 epochs, a learning rate of 10^{-4}, and a weight decay of 10^{-4}.

All the values proposed have been obtained after an exhaustive experimentation.

4 Experiments

We test the model proposed in all the challenges posed in Sec. 1. To this end split, the dataset in 80–20%, using 6400 *raw* strain matrices for training and 1600 for test. Each *raw* strain matrix undergoes its own data representation process.

4.1 Reconstruction. First, we incorporate RIM and the extra terms of the loss function L to the models in Table 1. The new comparison is shown in Table 3, in which each column contains the mean value of the loss represented in it. The down-arrow indicates that the lower the value, the better.

Since RIM matches the U-net, a key question arises: Is it truly necessary the new model? The following experiments show that our proposal completely overcomes the U-net. We also present a qualitative comparison between the reconstruction and the

Table 2 Values of the hyperparameters in RIM

Symbol	Meaning	Value
T	Number of iterations	5
γ	Multiplicative factor in enhancement function (Eq. (7))	0.03
τ	Time-step that activates enhancement function (Eq. (7))	3
ζ	Multiplicative factor in smoothing function (Eq. (8))	0.2
ν	Threshold for computing the mask matrix M in (Eq. (12))	0.3
$\lambda_0, \lambda_1, \lambda_2$	Weights of MSE, L_1 and L_2 in the loss function (Eq. (16))	1, 10, 1

Table 3 Comparison of state-of-the-art models with respect to ours, detailing each term of the loss L for the reconstruction task

Model	MSE (↓)	L_1(↓)	L_2(↓)	L(↓)
Ours	**0.002**	**0.004**	**0.007**	**0.014**
U-net	**0.002**	**0.004**	**0.007**	**0.014**
BVAE	0.016	0.017	0.017	0.051
CGAN	0.035	0.030	0.055	0.121

Note: Bold highlights the best value.

ground truth of ten random test instances in Fig. 2. It can be appreciated that in cases with a single load such as Figs. 2(*a*), 2(*d*), 2(*e*), and 2(*j*), the prediction is almost perfect. In the rest of the instances, with two or more loads, the prediction always hits at least half of the loads.

4.2 Robustness to Sensor Failure. Sensor failure is common if the structure is subject to impacts, such as in fighter fuselages or warship hulls, hence, to be able of reconstructing the strain image when some of the sensors readings are missing is a key issue. To this end, we configure nine failure patterns, namely, (0,1), (10,11), (0,10), (1,11), (2,3,4), (4,6,8), (7,8,9), (3,5,7), and (0,1,10,11), where each sensor is represented by its id as stated in Fig. 1(*b*).

Given the outcome in reconstruction, we focus only on the U-net and our proposal. The comparison is carried out averaging the value of L over all the test set. Then we obtain the percentage of times that each model attains the lowest value. The results are shown in Table 4. RIM overcomes the U-net in all the failure patterns but one, but precisely the one with the lowest difference.

4.3 Transferability. We here consider whether the model learned for a given morphology works in another one, without retraining nor fine-tuning. To this end, we consider other four morphologies with different number of sensors, namely, 8, 9, 13, and 16. The morphologies with 8 and 16 sensors are shown in Figs. 3(*a*) and 3(*b*), respectively. The morphology with 12 sensors is the original one, shown in Fig. 1(*b*). The morphologies with 9 and 13 sensors are identical to 8 and 12 but with an extra sensor in the middle. As in the previous experiment, we average the value of L over all the test set and obtain the percentage of times that each model attains the lowest value. The results are shown in Table 5, in which our model overcomes the U-net in all the cases.

4.4 Ablation of R, E, and S. Here, we assess the utility of functions R, E, and S. To this end, we consider RIM with and without R, E, and S and repeat the robustness and transferability experiments. The rest of the procedure is identical to those described earlier.

The results for the robustness ablation test are shown in Table 6. The original model overcomes the one without R, E, and S in seven out of nine cases. On the other hand, the results for the transferability ablation test are shown in Table 7. Here, model overcomes the one without R, E, and S in four of five cases.

4.5 Generalization to Three-Dimensional Hollow Structures. We also consider the case in which the geometry is not a plate, but a 3D thin-walled hollow structure, as in lattice-structured cubes used in lightweight construction. To this end, a new geometry was developed and simulated. Specifically, we try a cube with 10×10 elements in each one of its six faces, subject to random loads at any of them with the same protocol than in the thin-walled plate case. Since the number of elements per face is ten times lower, we modify the morphology, reducing the number of FBG sensors to 8. Finally, this 3D geometry is unwrapped into a 2D surface, so RIM can be readily applied. We also create 8000 synthetic strain matrices with FEM simulation and the same protocol than before. The geometry and morphology of this setup are shown in Figs. 4(*a*) and 4(*b*).

We compare the performance of a U-net with respect to our model under three metrics: the mean absolute error (MAE), the MSE, and the peak signal-to-noise ratio (PSNR). These metrics are computed for each image in the test set and then averaged. The results are shown in Table 8, and RIM is superior to the U-net in the three metrics.

4.6 Discussion on Real-World Validation. Real-world validation is a key issue that was considered during this work.

First, it is not feasible to conduct an experiment in which a plate is equipped with FBGs so densely that strain readings equivalent to the size of a finite element can be obtained for each region.

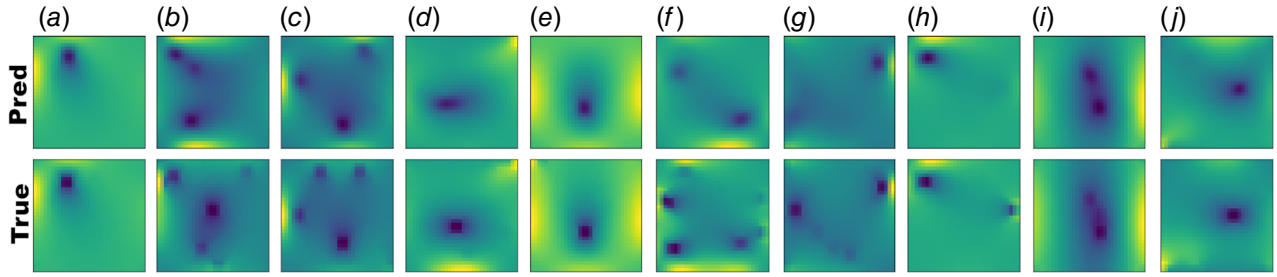

Fig. 2 Inpanited versus the ground truth image for ten random test instances

Table 4 Percentage of times that each model attains the lowest L when removing some sensors in test

	Id. of those sensors removed								
Model	0,1	10,11	0,10	1,11	2,3,4	4,6,8	7,8,9	3,5,7	0,1,10,11
U-net (↑)	32.7	1.8	3.6	21.8	36.4	14.6	**52.7**	38.2	0
Ours(↑)	**67.3**	**98.2**	**96.4**	**78.2**	**63.6**	**85.4**	47.3	**61.8**	**100**

Note: Bold highlights the best value.

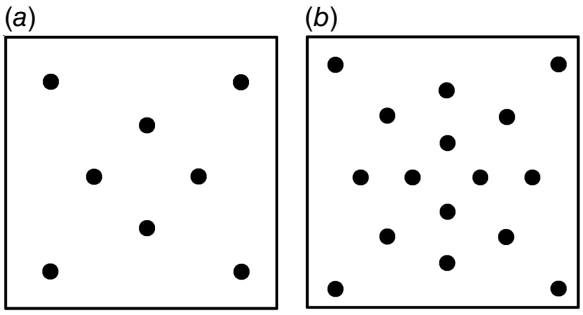

Fig. 3 Morphologies with 8 and 16 sensors

Table 5 Percentage of times that each model attains the lowest *L* in test using other morphologies without retraining or fine-tuning RIM

	No. of sensors in new morphology				
Model	8	9	12[a]	13	16
U-net (↑)	12	0	20	20	21
Ours (↑)	**88**	**100**	**80**	**80**	**79**

[a]The morphology learned with RIM.
Note: Bold highlights the best value.

Table 6 Percentage of times that RIM attains the lowest *L* in test not using functions *R*, *E*, and *S* compared to the original one within the robustness test

	Id. of those sensors removed								
Model	0,1	10,11	0,10	1,11	2,3,4	4,6,8	7,9,8	3,5,7	0,1,10,11
Ours (↑)	**65**	10	27	**89**	**78**	**92**	**55**	**75**	**85**
No R-E-S (↑)	35	**90**	**63**	11	22	8	45	25	15

Note: Bold highlights the best value.

Table 7 Percentage of times that RIM attains the lowest *L* in test not using functions *R*, *E*, and *S* compared to the original one within the transferability test

	No. of sensors in new morphology				
Model	8	9	12[a]	13	16
Ours (↑)	**100**	**100**	**55**	34	**100**
No R-E-S (↑)	0	0	45	**66**	0

[a]The morphology learned with RIM.
Note: Bold highlights the best value.

Table 8 Comparison of a U-net versus RIM on the reconstruction task

Model	MSE (↓)	MAE (↓)	PSNR (↑)
U-net	0.0005	0.007	33.6
Ours	**0.0002**	**0.005**	**38.2**

Note: Bold highlights the best value.

Reducing the number of sensors to 4 fibers and 12 gratings each results in an estimated cost of EUR 21,000 and 400 working hours split as follows:

Amount (EUR)	Description
1000	Manufacture of two[a] composite panels.
2000	Instrumenting two[a] panels with 4 fibers and 12 gratings each.
2000	Fastening elements for testing.
16,000	400 h of work (40 EUR/h) that include designing the experiments, manufacture, install, and assembly sensors in the structure, develop the test plan, carry out the tests, and obtain and validate results.

[a]Two panels would be the minimum required to see repetitiveness and have a back-up.

In our opinion, this cost is excessive, in terms of both working hours and financial investment, particularly considering that the purpose of finite element simulations is precisely to avoid such expenses.

An alternative would be to employ a 12-sensor morphology, as shown in Fig. 1(*a*), and evaluate the accuracy of predicting the strain in one sensor based on the others, similar to the experiments assessing robustness to sensor failure (Sec. 4.2). In this case, the instrumentation cost may be lower, but the remaining expenses would remain largely unchanged.

In summary, this work presents five different experiments leveraging finite element simulation, which provides reduced costs and risks, as well as flexibility in the experiment design and parallel execution of different scenarios. By contrast, conducting experiments on a physical test bench is significantly more expensive in terms of both time and budget.

5 Related Works

5.1 Inpainting. In this article, we approach the interpolation of strain matrices from a reduced number of sensors as the inpainting of an image. The most recent works in this computer vision task leverage diffusion models in the latent space [18,19]. These works, as other state-of-the-art inpainting models with outstanding performance [20–22], share common features that are severe drawbacks for the problem addressed here. (1) The size of these models is way

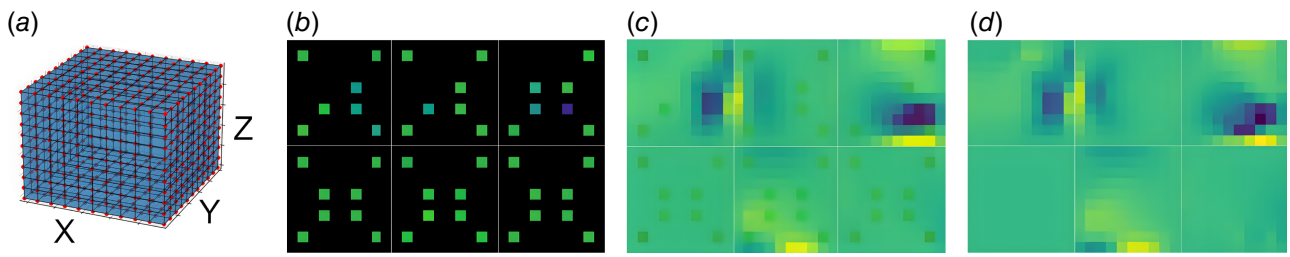

Fig. 4 (*a*) Geometry of the 3D hollow thin-walled structure used. (*b*) Unwrapping of the six sides of the 3D cube into a 2D plate, with the FBG readings from an eight-sensor morphology at each side. (c) Prediction and (*d*) target images.

larger than ours. (2) These models are trained with everyday images. Although coming from very different contexts, they induce a bias in the model's layers. Hence, transfer learning is not really useful for the images generated here, which leads to completely retrain the models from scratch. (3) The area missing in these works is much less than in this problem. (4) Besides, such an area usually contains a part with semantic meaning within the whole image (e.g., the face in a close-up, a person in a crowd or a tool in a hand), which does not happen here.

5.1.1 Estimation of Strain. This topic is very dependent on the test bench. For this reason, it is relevant to distinguish between neural network architectures and models, as the architectures provide guides to build promising models that must be tailored to the specific problem. Such is the case in Ref. [23], in which the U-net was proposed as a prominent architecture. In Ref. [24], the authors used a single ellipsoidal void structure as test bench. In contrast, we use a thin-walled plate and, as a special case in the experimentation, a hollow cube, which would be the closest structure to the one in that work.

Probably the best match with our study can be found in Ref. [25]. Here, the authors explored a deep learning-based approach for predicting the stress fields in 2D linear elastic cantilevered structures subjected to external static loads. The models tested include two models, SCSNet and StressNet, both based on the autoencoder architecture, while our model is also tested against other generative neural networks. Both Ref. [25] and our model are tailored for the test bench they are tested, so a quantitative comparison is not really possible. However, the U-net built as baseline in our article approximates the SCSNet but improves the architecture with the skip connections. Additionally, the test bench that we create is way larger than that presented by them.

5.1.2 Other Related Fields. Predictive maintenance is a comprehensive framework that aims to anticipate failures before they occur, using real-time data and degradation models. It encompasses two broad nonexclusive paradigms, namely, structural health monitoring (SHM) [26] and on-condition maintenance [27]. The scope of the former is to assess the global integrity and long-term health of large-scale structures and infrastructures, e.g., in Ref. [28], whereas the latter's is to maintain the operational performance of individual machinery, components, or assets by reacting to their current state. However, both overlap in three key aspects: (1) The use of vibration sensors, fiber optics, accelerometers, and extensometers, to name just a few, to acquire data indicative of an asset's or structure's state such as vibration analysis, acoustic emission, temperature; (2) both shift from time based or reactive maintenance to data driven, so actions are due to the actual condition of the asset or structure; (3) SHM focuses on long-term prognostics, so it is predictive in nature. The more advanced forms of on-condition maintenance also incorporate predictive capabilities using the same techniques such as machine/deep learning [29,30] or anomaly detection [31]. Digital twins are arguably the most comprehensive approach to health monitoring for they integrate sensoring, cyperphysical systems, machine learning models, and real-time actions [32]

Novel machine learning models, such as physics-informed neural networks [33] have emerged as a potential approach to integrating physical constraints into the learning process [34]. This approach has been used to model the behavior of materials with damage in Ref. [35]. However, their applicability is often limited to single-case scenarios under well-defined boundary conditions. Regrettably, a reduced number of sensors prevents the direct application of equilibrium equations for stress analysis, and additional stress-related measurements would be required for a complete characterization of structural behavior.

6 Conclusions

In this article, we turn an interpolation/reconstruction problem into the inpainting of an image with 98.8% of the pixels missing. To this end, we introduce RIM. We first create a massive synthetic dataset using FEM simulations of the test bench, creating strain matrices in both axis. Then propose a data representation that turns these strain matrices into images. We tailor state-of-the-art models as baseline for reconstruction (i.e., inpainting from the *image* point of view) and identify other challenges that the model should be able to cope with our model. Extensive experiments have been carried out to assessing the performance of RIM with respect to the best baseline, the U-net. RIM is superior in all the tasks, keeping a reduced number of parameters in comparison with other state-of-the-art inpainting models.

On the other hand, finite element–based simulation cannot fully capture real-world behavior once the material reaches critical strain levels, when cracking and fracturing occur. Hybrid approaches based on previous theoretical and experimental results are typically required to ensure reliability in such extreme conditions. In that sense, simulations produced by RIM can predict whether those critical bounds are reached.

Author Contribution Statement. A. Cruz-Alonso: Investigation, formal analysis, software, conceptualization, methodology, writing – original draft. F. Terroba: Resources, funding acquisition, supervision, project administration. A. Cuesta-Infante: Writing – review and editing, visualization, supervision, project administration.

Funding Data

- The Spanish Government (Ministerio de Ciencia e Innovación) (Contracts PID2021-127879OA-C22 (CRYFTO: TowaRds the design of saFe liquid hYdrogen tanks for effiCient and green Transport applicatiOns)).
- R&D Project TED2021-129162B-C22 (MICIU/AEI/10.13039/501100011033/).
- The European Union NextGenerationEU/PRTR.
- R&D Project PID2021-128362OB-I00 (MICIU/AEI/10.13039/ 501100011033/ and FEDER/UE).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The data and information that support the findings of this article are freely available at: https://github.com/ancral/rim.

References

[1] Li, Y., Wen, C.-Y., Liu, X., Zhang, W., and Zheng, Y., 2025, "Prescribed-Time Fault-Tolerant Flight Control for Aircraft Subject to Structural Damage," IEEE Trans. Aeros. Electronic Syst., **61**(2), pp. 1848–1859.

[2] Wang, M., Gao, W., Wang, F., Zhao, X., and Zhang, J., 2025, "Theoretical, Numerical, and Experimental Analyses of the Buckling of Thin-Walled Sandwich Cylindrical Shells Subjected to External Pressure," Ocean Eng., **327**, p. 120950.

[3] Castanie, B., Azoti, W., Crouzeix, L., Bello, A., Taborda, R. P., Mahmood, A., and Viste, A., 2025, "Review of Monolithic Composite Laminate and Stiffened Structures in Aeronautic Applications," Composites Part C: Open Access, **17**, p. 100585.

[4] Keller, T., Zand, S. M., and Habibi, T., 2025, "Composite Engineering Structures in Seawater—Review of Durability and Environmental Performance," Compos. Struct., **357**, p. 118891.

[5] Lin, Y., Qian, W., Lei, L., Liu, Y., Zhang, J., Liu, J., Kong, W., Hu, Y., Shi, Y., Wu, Z., Liu, H., and Wu, S., 2025, "Structural Integrity Issues of Composite Materials and Structures in Future Transportation Equipment," Compos. Struct., **358**, p. 118943.

[6] Alaswad, S., and Xiang, Y., 2017, "A Review on Condition-Based Maintenance Optimization Models for Stochastically Deteriorating System," Reliab. Eng. Syst. Saf., **157**, pp. 54–63.
[7] Meemary, B., Vasiukov, D., Deléglise-Lagardére, M., and Chaki, S., 2025, "Sensors Integration for Structural Health Monitoring in Composite Pressure Vessels: A Review," Compos. Struct., **351**, p. 118546.
[8] Wang, B., Zhong, S., Lee, T.-L., Fancey, K. S., and Mi, J., 2020, "Non-Destructive Testing and Evaluation of Composite Materials/structures: A State-of-the-Art Review," Adv. Mech. Eng., **12**(4), p. 1687814020913761.
[9] Deng, Y., Zhao, Y., Ju, H., Yi, T.-H., and Li, A., 2024, "Abnormal Data Detection for Structural Health Monitoring: State-of-the-Art Review," Develop. Built Environ., **17**, p. 100337.
[10] Zheng, W., Li, J., Hao, H., and Fan, G., 2025, "Missing Data Imputation for Structural Health Monitoring Using Unsupervised Domain Adaptation and Pretraining Techniques," Eng. Struct., **328**, p. 119694.
[11] Bathe, K.-J., 2014, *Finite Element Procedures*, 2nd ed., Prentice Hall, Upper Saddle River, NJ.
[12] Mesquita, E., Antunes, P., and Coelho, F. e. a., 2016, "Global Overview on Advances in Structural Health Monitoring Platforms," J. Civil Struct. Health Monit., **6**, pp. 461–475.
[13] Kromanis, R., and Kripakaran, P., 2014, "Predicting Thermal Response of Bridges Using Regression Models Derived From Measurement Histories," Comput. Struct., **136**, pp. 64–77.
[14] Ronneberger, O., Fischer, P., and Brox, T., 2015, "U-Net: Convolutional Networks for Biomedical Image Segmentation," Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, Oct. 5–9, Springer International Publishing, pp. 234–241.
[15] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A., 2017, "beta-VAE: Learning Basic Visual Concepts With a Constrained Variational Framework," International Conference on Learning Representations, Toulon, France, Apr. 24–26.
[16] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., 2017, "Image-to-Image Translation With Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, July 21–26, pp. 5967–5976.
[17] Ho, J., Jain, A., and Abbeel, P., 2020, "Denoising Diffusion Probabilistic Models," *Advances in Neural Information Processing Systems*, H. Larochelle, M. A. Ranzato, R. Hadsell, M. F. Balcan and H. T. Lin, eds., Vol. 33, pp. 6840–6851.
[18] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., 2022, "High-Resolution Image Synthesis With Latent Diffusion Models," IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, June 18–24, pp. 10674–10685.
[19] Corneanu, C., Gadde, R., and Martinez, A. M., 2024, "Latentpaint: Image Inpainting in Latent Space With Diffusion Models," IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, Jan. 4–8, pp. 4322–4331.
[20] Deng, Y., Hui, S., Meng, R., Zhou, S., and Wang, J., 2022, "Hourglass Attention Network for Image Inpainting," 17th European Conference on Computer Vision (ECCV), Milan, Italy, Sept. 29–Oct. 4, Vol. 13678, pp. 483–501.
[21] Zhuang, J., Zeng, Y., Liu, W., Yuan, C., and Chen, K., 2023, "A Task Is Worth One Word: Learning With Task Prompts for High-Quality Versatile Image Inpainting," European Conference on Computer Vision (ECCV), Springer-Verlag, pp. 195–211.
[22] Ko, K., and Kim, C.-S., 2023, "Continuously Masked Transformer for Image Inpainting," IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, Oct. 2–6, pp. 13123–13132.
[23] Bhaduri, A., Gupta, A., and Graham-Brady, L., 2022, "Stress Field Prediction in Fiber-Reinforced Composite Materials Using a Deep Learning Approach," Compos. Part B: Eng., **238**(6), p. 109879.
[24] Bhaduri, A., Ramachandra, N., Krishnan Ravi, S., Luan, L., Pandita, P., Balaprakash, P., Anitescu, M., Sun, C., and Wang, L., 2024, "Efficient Mapping Between Void Shapes and Stress Fields Using Deep Convolutional Neural Networks With Sparse Data," J. Comput. Inf. Sci. Eng., **24**(5), p. 051008.
[25] Nie, Z., Jiang, H., and Kara, L. B., 2019, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," ASME J. Comput. Inf. Sci. Eng., **20**(1), p. 011002.
[26] Huang, X., Wang, P., Zhang, S., Zhao, X., and Zhang, Y., 2022, "Structural Health Monitoring and Material Safety With Multispectral Technique: A Review," J. Safety Sci. Resilience, **3**, pp. 48–60.
[27] Ahmed Murtaza, A., Saher, A., Hamza Zafar, M., Kumayl Raza Moosavi, S., Faisal Aftab, M., and Sanfilippo, F., 2024, "Paradigm Shift for Predictive Maintenance and Condition Monitoring From Industry 4.0 to Industry 5.0: A Systematic Review, Challenges and Case Study," Res. Eng., **24**, p. 102935.
[28] Contreras Lopez, J., Chiachío, J., Saleh, A., Chiachío, M., and Kolios, A., 2022, "A Cross-Sectoral Review of the Current and Potential Maintenance Strategies for Composite Structures," SN Appl. Sci., **4**(6), pp. 2523–3971.
[29] Oh, B., Glisic, B., Kim, Y., and Park, H., 2020, "Convolutional Neural Network-Based Data Recovery Method for Structural Health Monitoring," Struct. Health. Monit., **19**, pp. 1821–1838.
[30] Malekloo, A., Ozer, E., AlHamaydeh, M., and Girolami, M., 2022, "Machine Learning and Structural Health Monitoring Overview With Emerging Technology and High-Dimensional Data Source Highlights," Struct. Health. Monit., **21**, pp. 1906–1955.
[31] Quatrini, E., Costantino, F., Di Gravio, G., and Patriarca, R., 2020, "Machine Learning for Anomaly Detection and Process Phase Classification to Improve Safety and Maintenance Activities," J. Manuf. Syst., **56**, pp. 117–132.
[32] Karkaria, V., Chen, J., Luey, C., Siuta, C., Lim, D., Radulescu, R., and Chen, W., 2025, "A Digital Twin Framework Utilizing Machine Learning for Robust Predictive Maintenance: Enhancing Tire Health Monitoring," ASME J. Comput. Inf. Sci. Eng., **25**(7), p. 071003.
[33] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2019, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," J. Comput. Phys., **378**, pp. 686–707.
[34] Niu, S., Zhang, E., Bazilevs, Y., and Srivastava, V., 2023, "Modeling Finite-Strain Plasticity Using Physics-Informed Neural Network and Assessment of the Network Performance," J. Mech. Phys. Solids, **172**, p. 105177.
[35] Janssen, J. A., Haikal, G., DeCarlo, E. C., Hartnett, M. J., and Kirby, M. L., 2024, "A Physics-Informed General Convolutional Network for the Computational Modeling of Materials With Damage," ASME J. Comput. Inf. Sci. Eng., **24**(11), p. 111002.