Mechanical Systems and Signal Processing 232 (2025) 112743

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

Full Length Article

Data-driven stochastic model updating and damage detection with deep generative model

Tairan Wang [a], Sifeng Bi [a,*], Yanlin Zhao [b], Laurent Dinh [c], John Mottershead [d]

[a] Department of Aeronautics and Astronautics, University of Southampton, UK
[b] School of Mechanical Engineering, University of Science and Technology Beijing, China
[c] Apple, Cupertino, CA, United States
[d] Department of Mechanical and Aerospace Engineering, University of Liverpool, UK

ARTICLE INFO

Keywords:
Model updating
Uncertainty quantification
Deep generative model
Invertible neural network
Conditional neural network
Structural health monitoring

ABSTRACT

Is there a calibration algorithm beyond the dominant Bayesian sampling approach and sensitivity-based optimisation in model updating? Can a neural network serve not only as a surrogate model but also possess its own calibration capacity, independent of the Bayesian or optimisation framework? This work aims to address these questions by developing a unique data-driven approach for stochastic model updating and damage detection. Among a variety of models in deep learning, the class of deep generative model shares a similar objective, to estimate an unknown or intractable probability distribution from a small number of samples, with model updating. As a powerful flow-based deep generative model, a recently developed conditional Invertible Neural Network (cINN) architecture has been adopted in the task of model updating. Unlike the conventional approaches that employ the neural networks solely as a forward surrogate, the cINN-based model updating is a framework that performs as a bidirectional network where the forward training and inverse calibration are integrated into a uniform structure. The cINN consists of two parts known as the conditional network and the invertible neural network (INN). Both networks are trained jointly in the forward direction and can operate inversely to offer rapid and accurate predictions by given observation data. The application of the cINN provides a more efficient and direct manner to solve model updating problems without calculating the likelihood function in Bayesian inference. The cINN is embedded into a multilevel stochastic updating framework. Rather than directly calibrating physical parameters, this multilevel framework focuses on their statistical moments, e.g. mean and variance, referred to as hyperparameters. The hyperparameters are then utilised to determine the probability of damage (PoD), which provides a confidence level about the structural condition, facilitating stochastic damage detection. Two case studies are proposed to demonstrate the multilevel cINN-based stochastic updating and damage detection approach. The first involves a 3-degree-of-freedom spring-mass simulation model, while the second case study employs an experimental rig test-case with practical measurements, each under various damage scenarios.

* Corresponding author.
E-mail address: sifeng.bi@soton.ac.uk (S. Bi).

https://doi.org/10.1016/j.ymssp.2025.112743
Received 3 November 2024; Received in revised form 1 April 2025; Accepted 10 April 2025
Available online 23 April 2025

1. Introduction

As a process to identify damage occurrence for in-service engineering structures, Structural Health Monitoring (SHM) has become a crucial technique to ensure the safety, reliability, and serviceability of engineering infrastructures [1]. A well-developed SHM system tends to detect anomalies as early as possible and carry out timely repair and maintenance actions [2]. Damage detection has been investigated and conducted by various methods such as vibration-based testing [3], acoustic emission testing [4], computer-vision-based testing [5], and data-driven machine learning testing [6], etc. Among these various methods, the practice of model updating plays a crucial role in SHM since its remarkable ability to identify any changes of the structure parameters, or their uncertainty characteristics, given the available measurement data from the structures in-serve [7,8].

The model updating technique involves adjusting physical or non-physical parameters in the computational model to enhance its agreement with experimental results [9]. The typical model updating was carried out in the deterministic domain where the sensitivity-based method [10] was proved to be the most effective for solving deterministic model updating problems. However, due to the presence of hybrid uncertainties in both numerical models and physical experiments, it is necessary to extend beyond deterministic single-simulation-single-test scenarios to capture the variability in the simulation and experimental data [10,11]. Thus, the stochastic model updating technique is then developed to overcome these drawbacks [12,13]. The famous Bayesian model updating framework was proposed by Beck and Katafygiotis [14] as a fundamental tool for stochastic model updating and further developed by Beck and Au [15] by implementing the Markov Chain Monte Carlo (MCMC) algorithm in this framework.

The Bayesian inference is widely adopted as a powerful method to solve the inverse parameter calibration problem. It is derived by obtaining the posterior distribution of the uncertain model parameters based on prior knowledge (prior distribution) and the likelihood function, which describes the probability of the observation data given different values of the model parameters [16,17], with the aid of numerical sampling algorithms. The determination of the likelihood function is the most essential segment in the Bayesian model updating framework. Nevertheless, in many instances, the establishment of the likelihood function is always either computationally expensive due to the high-dimensional integral calculations and the quantification of hybrid uncertainties, or analytically intractable because of the complexity of the model. To address this computation challenge, the approximate Bayesian computation (ABC) is proposed, where the original likelihood definition is replaced by an approximate but effective function. The Bhattacharyya distance-based approximate likelihood function is proven to be viable with its remarkable ability to capture discrepancies between numerical simulations and experimental observations by calculating the overlap between two distributions [18,19]. Some sampling methods [20] such as MCMC and Transitional MCMC (TMCMC) [21] are also utilised to sample from the posterior distributions of calibrating parameters to estimate their true value without evaluating the normalising factor (i.e., evidence) in the Bayes' Theorem.

The damage detection technique was classified as a four-level paradigm [22] including (1) damage identification; (2) damage localisation; (3) damage severity assessment; and (4) remaining lifetime prediction. The Bayesian model updating-based damage detection methods can provide a promising function for the first three levels by estimating the posterior probability distribution for the structural parameters [23]. Although the ABC is already an effective tool for solving model updating problems, it still faces some problems like over-reliance on the accuracy of sampling methods and is time-consuming when handling some high-dimensional tasks that cannot fulfil the timeliness and accuracy required in SHM [24]. It is then the tendency to utilise data-driven techniques to facilitate model updating and damage detection. However, most existing data-driven approaches still focus on developing surrogates to replace the time-consuming finite element model. These approaches follow a similar style as the typical surrogate modelling approach but use various machine learning models, such as artificial neural networks (ANN) and Bayesian neural networks (BNN), to replace conventional response surface or Kriging models. Nevertheless, since the machine learning models still fall within the Bayesian updating interface, the computationally expensive likelihood evaluation is still unavoidable. Furthermore, conventional ANN or BNN models lose the physical significance of actual structural parameters, making them difficult to apply in physics-based structural health monitoring.

Apart from the conventional neural networks, a group of deep learning algorithms called deep generative models (e.g., Variational Autoencoders (VAEs) [25], Generative Adversarial Networks (GANs) [26], or Flow-based Models [27–29]) are widely used in various fields such as text analysis, image analysis, medical imaging, etc [30]. They also show the potential to solve inverse problems like model updating since they have the ability to generate new data or simulate possible results that can reflect the true data distribution of the real-world system. The recently developed conditional invertible neural network (cINN) [31,32] is a flow-based deep generative model developed with a unique invertible architecture, allowing for the integration of physical observations and posterior estimation. The technique introduces the potential to solve the above-mentioned challenges regarding the data-driven approach in stochastic model updating and damage detection.

The cINN was developed and successfully applied to solve guided image generation problems in the field of computer vision [31]. It was also implemented to address the inverse problems [33,34], damage detection problems [35], and wind turbine design [36] that demonstrate the potential of cINN in engineering. The architecture of cINN consists of an invertible neural network (INN) and a conditional network, where the conditional network is responsible for reducing a set of observations of variable size to a fixed-size vector of learned summary statistics. The INN is accountable for mapping input data to the latent space in the training phase and predicting the true posterior of model parameters given the summary statistics of the observed data as the condition in the inference phase.

The cINN function is a bijective model that establishes bidirectional mappings between a complex posterior distribution, the prior distribution, and the observations, directly bypassing the need for likelihood evaluation as required in Bayesian inference. This is achieved by jointly training the INN and the conditional network in the forward direction, using both model inputs and outputs as training data. The transformation from the complex, unknown probabilistic distribution to a simple distribution, referred to as the

latent distribution, is based on the Change of Variable Rule (CoVR). The cINN is trained by minimising the Kullback-Leibler (KL) divergence between the true target distribution and the learned posterior distribution. Once trained, the cINN can operate inversely by sampling from the latent distribution, conditioned on the observation data, to generate the corresponding input samples and derive the posterior distribution. The invertibility of the cINN is guaranteed by the special architecture of the Affine Coupling Layer (ACL) [28], which simplifies the execution of the CoVR by reducing the complexity of evaluating the Jacobian determinant.

The latent distribution plays a crucial role in cINNs, as it maps data from the model's input space to the latent variable space, conditioned on additional information such as model outputs or practical observation data. In cINN-based model updating, the main objective is to infer input data based on practical observations. The conditional network, together with the INN, recovers the complex distribution of input parameters from sparse and noisy observation data. By conditioning the observed data, the latent distribution enables the model to produce a distribution over possible inputs, capturing the uncertainty and variability in the inverse mapping. The introduction of the latent distribution helps avoid a many-to-one relationship between input and observation data, where an element of input data is connected to a combination of observation data and latent space elements.

In this manuscript, we provide a detailed explanation of the structure of the ACL, followed by the design of two coupled ACL units, which together establish the invertible bijective mapping between the original distribution and the latent distribution. The principles behind the CoVR, as well as the method for evaluating the Jacobian determinant within the ACL architecture, are thoroughly discussed. Additionally, a detailed explanation is provided on how to apply the cINN to typical model updating tasks. During the forward training phase, input parameter samples are drawn from a prior distribution, serving as the initial distribution to be transformed into a pre-selected latent distribution (commonly a standard Gaussian distribution). Simultaneously, model output samples are passed through the conditional network, ensuring that the transformation from the prior distribution to the latent distribution captures the necessary information from the numerical model. In the inverse inference phase, random samples are generated from the simple latent distribution. The corresponding input parameter samples are inferred by passing the available observation data through the conditional network to yield the posterior distribution.

The cINN-based forward training and inverse inference process integrates seamlessly into the stochastic model updating procedure, generating a posterior distribution of possible input parameters conditioned on the available observation data. The posterior distribution of structural parameters is subsequently applied in damage detection, where the Probability of Damage (PoD) is evaluated to assess the structural condition.

The rest of the paper is organised as follows. The fundamentals of stochastic model updating and the limitations of the Bayesian approach are presented in Section 2. The structure and the features of the cINN, and the framework of cINN-based model updating are introduced in Section 3. In Section 4, the process of cINN-based model updating for stochastic damage identification and the definition of probability of damage (PoD) are introduced in detail. Two case studies including a simulation-based 3-DOF spring-mass system and an experimental rig structure with controllable uncertainties are presented in Section 5 and Section 6 to demonstrate the effectiveness and capacity of the proposed approach. A comparison of the accuracy and efficiency of the cINN-based method with the sensitivity method and Bayesian method is demonstrated at the end of Section 5 & 6. Section 7 gives the conclusions and perspectives.

2. Stochastic model updating: Bayesian approach and its limitation

2.1. Numerical model, parameters, and uncertainties

The stochastic model updating problem generally involves a physical system that is modelled numerically by a function $y_{sim} = M(x)$ whereby x represents the vector of the input parameters, y_{sim} represents the simulation output features, and $M(\cdot)$ stands for the simulator. The simulator can be performed as a sophisticated Finite Element (FE) model or a simplified mathematical function. Normally, this simulator may describe both static and dynamic situations and can be either linear or non-linear [20].

The mathematical relation between the experimental observation y_{obs} and the numerical simulation y_{sim} can be expressed as:

$$y_{obs} = y_{sim} + \epsilon = M(x) + \epsilon \tag{1}$$

where ϵ represents the discrepancy between the numerical simulation and the experimental observation caused by the existence of uncertainty in both model parameters and measurements. The objective of stochastic model updating is to minimise the discrepancy so that the simulation results are in maximum agreement with the experimental measurements. Thus, the stochastic model updating is concluded as an inverse process that calibrates the uncertainty features (i.e. probabilistic distribution or interval) of input parameters by giving multiple observations. The Bayesian updating framework is a widely accepted approach for solving such problems.

2.2. Bayesian approach and its limitations

In the Bayesian multilevel model updating approach, the input parameters are totally determined by their probabilistic hyperparameters, i.e., the statistical moments such as mean and standard deviation ($\theta = [\mu, \sigma]$). In this way, the hyperparameters together with the assumed distribution format are able to determine the model input parameters, where the input parameters are aleatory and the hyperparameters that have distinct but unknown values are epistemic.

The Bayesian inference has the advantage of integrating the prior information and observation data to obtain the posterior. In the Bayes' formula:

$$P(\theta|y_{obs}) = \frac{P_L(y_{obs}|\theta) \cdot P\theta}{P(y_{obs})} \tag{2}$$

where $P(\theta)$ represents the prior distribution; $P_L(y_{obs}|\theta)$ refers to the likelihood function of the parameters; $P(y_{obs})$ represents the evidence (or normalising factor); $P(\theta|y_{obs})$ represents the posterior distribution. The prior distribution represents the initial hypothesis about the selected model parameters to be calibrated before any measurements. The likelihood function reflects the agreement between the output from experimental measurements and numerical simulations. The evidence/normalisation factor in the Bayesian formula ensures that the posterior distribution integrates to 1. However, evaluating this normalisation factor is challenging, as the analytical form of the likelihood function is often difficult to obtain. To address this, popular techniques such as MCMC have been developed, allowing the estimation of posterior distributions without explicitly computing the normalisation factor. Instead, these methods rely on an iterative sampling process to approximate the posterior distribution efficiently. Consequently, the Eq. (2) can be simplified as:

$$P(\theta|y_{obs}) \propto P_L(y_{obs}|\theta) \cdot P(\theta) \tag{3}$$

The definition of the likelihood function is significant in model updating because it contains information on both observation data and uncertain parameters to be calibrated. The likelihood can be calculated as Eq. (4) while considering n_{exp} times of independent experimental observations.

$$P_L(y_{obs}|\theta) = \prod_{i=1}^{n_{exp}} P(y_{obs}^i|\theta) \tag{4}$$

To obtain the precise estimation of the explicit distribution of each measurement, a large number of model evaluations are required. The approximate likelihood function was proposed to reduce the computation burden, alternatively, which is determined by various distance-based UQ metrics expressed as Eq. (5).

$$P_L(y_{obs}|\theta) \propto \exp\left\{ -\frac{d\left(y_{exp}, y_{sim}\right)^2}{\varepsilon^2} \right\} \tag{5}$$

where $d\left(y_{exp}, y_{sim}\right)$ represents the distance-based UQ metrics such as Euclidean, Mahalanobis, and Bhattacharyya distance [18,37].

Another challenge in the Bayesian framework is the evaluation of the normalising factor $P(y_{obs})$ in Eq. (2). The evaluation of $P(y_{obs})$ requires integrating over the likelihood function to ensure that the posterior distribution is a valid Probability Density Function (PDF), meaning the total integral of the posterior distribution equals one. Even when using an approximate likelihood function in Eq. (5) to replace the complex original definition, obtaining this integral remains difficult. To circumvent this issue, sampling-based techniques like MCMC and TMCMC are commonly used. These methods bypass the need to calculate the normalising factor by generating samples directly from the unknown posterior distribution. The posterior is then estimated using these obtained samples, effectively avoiding the challenging integral computation. A comprehensive tutorial on the multilevel Bayesian updating with the above random sampling algorithms is presented in Ref [9] for the reader's further reference.

Even with the use of an approximate likelihood function and sampling methods like MCMC to handle the challenges of evaluating both the likelihood and the normalising factor, some significant issues remain. These methods often require a large number of model evaluations to achieve an accurate estimation of the likelihood function, and MCMC, in particular, can be computationally slow,

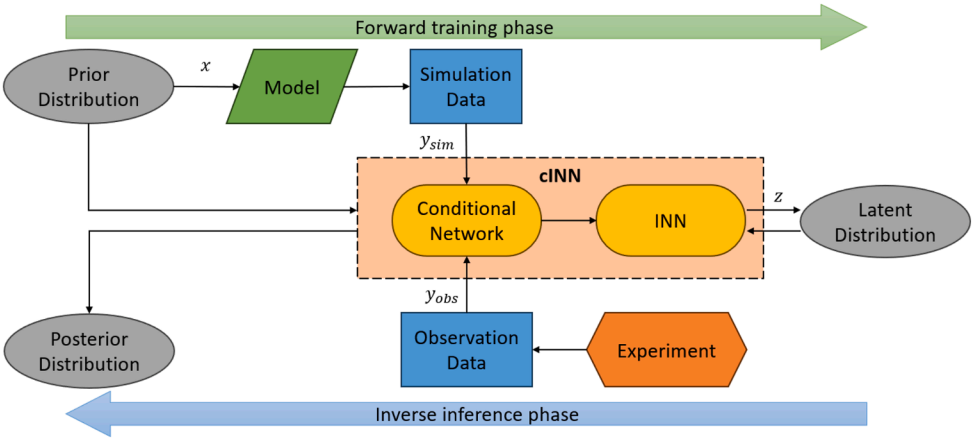


Fig. 1. The architecture of the cINN-based model updating.

especially when dealing with complex models or high-dimensional problems.

3. Model updating with conditional invertible neural networks (cINN)

Amongst various distinct deep generative models, cINN was adopted for model updating since it has a unique reversible structure, which allows for forward training and backward inference. The cINN-based model updating approach is capable of addressing the challenges mentioned above, offering a complete solution by entirely bypassing the need for likelihood evaluation and the normalising factor. The cINN framework directly generates posterior distributions through its bijective mapping mechanism, which allows for fast and accurate inference with fewer data requirements. By using the invertible neural network and conditional network architecture, cINN eliminates the need for complex sampling processes, providing a more efficient and direct alternative for stochastic model updating and damage detection.

3.1. Framework of cINN-based model updating

The cINN-based model updating framework is illustrated in Fig. 1, where the cINN is embedded into the bidirectional model updating process. This framework not only builds mappings from the input prior distribution to the latent space but also reverses this mapping to recover the posterior distribution of the input parameters, conditioned on the observation data. The cINN architecture combines an Invertible Neural Network (INN) with a conditional feed-forward network, both of which are trained jointly in an efficient manner.

In the forward training phase, the process starts by generating training data by sampling the input parameters from the prior distribution, which reflects initial assumptions or known distributions of the model parameters. These input samples are passed through the numerical model to generate the corresponding simulation outputs. The cINN's Conditional Network is responsible for processing the simulation outputs and conditioning the transformation, ensuring that the generated latent space encodes not only the information of the input data but also the simulation data. The processed conditional data is then injected into the INN for building up mappings between the latent distribution, which is typically a standard Gaussian distribution, and the input data conditioned on the summarised simulation data, y_{sim}. This forward training step is crucial as it trains the cINN to accurately capture and transform the complex distribution of input parameters into the latent space while retaining the information from the simulation outputs. The internal structure of the cINN, including how the conditional network and INN collaborate to perform this transformation, will be detailed in the following sections.

Once the cINN has been trained and the latent distribution is learned, the framework moves into the inverse inference phase. In this phase, the goal is to use the observation data to infer the posterior distribution of the input parameters. In this step, random samples are generated from the latent distribution (now known to follow a standard Gaussian) and passed back through the INN, conditioned on the summarised observation data using the conditional network. This inverse mapping allows the model to recover the corresponding input parameters, which are now informed by both the prior assumptions and the new observation data. The result is a refined posterior distribution of the input parameters, which better reflects the observation data.

3.2. Internal structure of cINN and the change of variable rule

This subsection provides a detailed explanation of the internal structure of the cINN and describes how the bijective mapping is achieved through its special architecture. As shown in Fig. 1, the cINN consists of two key components: the Conditional Network and the Invertible Neural Network (INN). The INN serves as the core element responsible for constructing a bijective transformation between the complex, irregular input distribution (prior or posterior), and a simplified latent distribution.

3.2.1. INN and the bijective mapping

The latent distribution plays a crucial role in cINNs, acting as a simplified representation of the input data. Typically modelled as a standard Gaussian distribution, the latent distribution is used to make the inverse process more tractable and computationally efficient. The transformation from input data to latent space allows for easier inference of the posterior distribution when conditioned on observation data. This latent space is where the complexities of the original input distribution are reduced, simplifying the learning and inference processes.

The transformation between the input distribution and the latent space is governed by the Change of Variable Rule (CoVR), a fundamental concept in probabilistic theory. CoVR is used to relate probability densities in one space to those in another when a transformation is applied. Specifically, it accounts for how the probability density changes as the variables are transformed, ensuring that the overall probability mass is preserved during the transformation. Mathematically, this involves the Jacobian determinant of the transformation, which adjusts for the changes in volume in the transformed space. The CoVR can be expressed in the following equation:

$$P_x(x) = P_Z(Z) \cdot \det\left(\frac{\partial f(x)}{\partial x}\right) = P_Z(f(x)) \cdot \det\left(\frac{\partial f(x)}{\partial x}\right) \tag{6}$$

In this equation, x is the model input parameters, and Z is the latent variable. $P_x(x)$ and $P_Z(Z)$ are the PDF of x and Z, where Z is typically assumed to follow a simple standard Gaussian distribution. The function $f(x)$ denotes the invertible transformation between x

and Z.

The transformation requires the calculation of the Jacobian determinant, $\det\left(\frac{\partial f(x)}{\partial x}\right)$, which consists of the partial derivatives of the transformation function concerning the input parameters. To ensure a valid mapping between the input parameters and the latent variables, the transformation must be bijective, meaning that it is both invertible and unique. Additionally, the transformation must be computationally efficient in both the forward and inverse directions, with a tractable Jacobian that allows for the explicit computation of the posterior probability. To meet these requirements, a special architecture known as the Affine Coupling Layer (ACL) [32] was designed, as shown in Fig. 2.

3.2.1.1. Forward direction. In the forward direction, as shown in Fig. 2(a), the D-dimension input vector x is split into two parts randomly: x_1 with d-dimension and x_2 with $(D-d)$-dimension. This splitting allows for flexibility in applying transformations to only part of the input, while leaving the other part unchanged. Specifically, x_1 remains unchanged throughout the forward process, so $z_1 = x_1$.

On the other hand, x_2 undergoes a transformation governed by two functions $s_1(x_1)$ and $t_1(x_1)$. These functions, typically represented by standard feedforward networks, take x_1 as input and output scaling and translation values for x_2. These scaling function $s_1(x_1)$ apply an element-wise scaling to x_2 through the operation $x_2 \odot \exp(s_1(x_1))$, where $\odot$ denotes the element-wise multiplication. The use of the exponential function ensures that the transformation remains invertible, as the exponential is always positive, making it possible to later reverse the operation. Additionally, a translation function $t_1(x_1)$ shifts x_2 by adding $t_1(x_1)$. Thus, the full transformation is given as

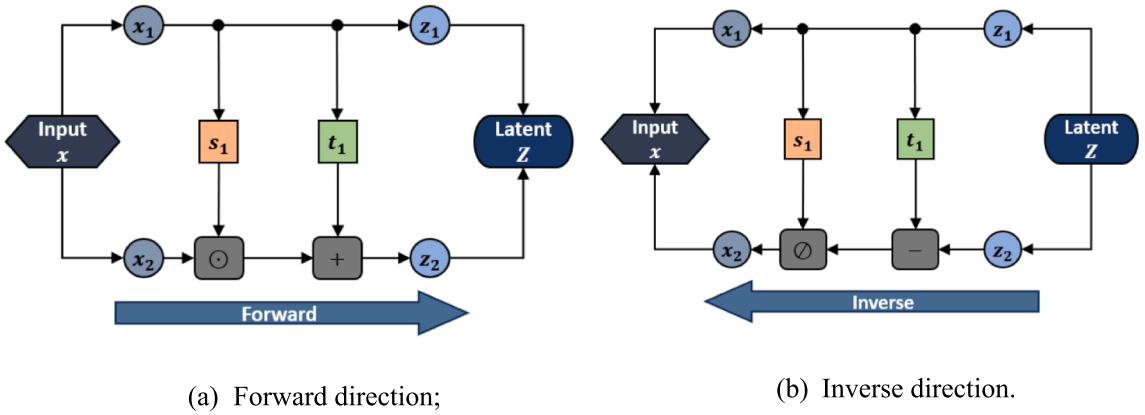
$$\begin{aligned} z_1 &= x_1 \\ z_2 &= x_2 \odot \exp(s_1(x_1)) + t_1(x_1) \end{aligned} \tag{7}$$

The scaling function $s_1(\bullet)$ and the translation function $t_1(\bullet)$, being neural networks, allow for a flexible and complex transformation that preserves the ability to reverse it. The outputs from the transformed latent variables, $Z = [x_1; z_2]$, which can be passed to the next layer.

3.2.1.2. Inverse direction. In the inverse direction, depicted in Fig. 2(b), the latent variables z_1 and z_2 are used to recover the original input vector x. Since x_1 was not altered in the forward transformation, it is directly retrieved from the latent variable as $x_1 = z_1$. However, recovering x_2 requires reversing the scaling and translation transformations applied during the forward process. The inverse transformation first undoes the translation by subtracting $t_1(z_1)$ from z_2. Then, the element-wise scaling is reversed by multiplying by the reciprocal of the exponential of $s_1(z_1)$, ensuring the scaling effect is undone. The full expression for recovering is:

$$\begin{aligned} x_1 &= z_1 \\ x_2 &= (z_2 - t_1(z_1)) \odot \exp(-s_1(z_1)) \end{aligned} \tag{8}$$

In addition, the special structure of the ACL enables an easy-to-compute Jacobian determinant, a critical component in the CoVR. The transformation's Jacobian matrix, which contains the partial derivatives of the transformed variables with respect to the input variables, is essential for determining how the probability density changes during the transformation. In the ACL, since x_1 remains unchanged and only x_2 is modified by $s_1(x_1)$, the Jacobian matrix becomes a simple diagonal matrix, where the non-zero elements are the exponential terms $\exp(s_1(x_1))$, expressed as



(a) Forward direction; (b) Inverse direction.

Fig. 2. The structure of the affine coupling layer (ACL).

$$J = \begin{bmatrix} \mathbb{I} & 0 \\ \dfrac{\partial z_2}{\partial x_1} & \mathrm{diag}(\exp(s_1(x_1))) \end{bmatrix} \tag{9}$$

This makes the determinant of the Jacobian matrix easily computable as:

$$\det(\boldsymbol{J}) = \exp\left(\sum s_1(x_1)\right) \tag{10}$$

This simplified computation allows for efficient calculation of the probability densities during both forward and inverse transformations, making the ACL architecture suitable for real-time inference and model updating tasks.

3.2.2. *The conditional network within cINN*

In the overall cINN-based framework, as shown in Fig. 3, the Conditional Network plays a crucial role in incorporating external information from the model outputs or observation data into the INN. This additional information, denoted by $\boldsymbol{y}_{sim}$ (simulation data) and $\boldsymbol{y}_{obs}$ (observation data), acts as a condition that modifies how the input parameters x are transformed into the latent variables Z, and vice versa.

The Conditional Network, represented by multiple layers in Fig. 3, takes the model's simulation outputs or the real observation data as input. It processes this information through a series of neural network layers to extract relevant features that are used to condition the transformations performed by the INN. The extracted features encode the relationship between the input parameters and the latent variables, ensuring that the transformation from x to the latent variables Z (and back) captures the necessary dependencies between the model and the real-world data.

In practical stochastic model updating tasks, the Conditional Network ensures that the mapping between input parameters and latent data is adaptive and informed by both the simulation outputs and the real-world observations. The network's ability to condition the transformation based on data from both sources enables it to address key challenges in model updating, such as dealing with sparse or noisy observation data and making the process robust to uncertainties. By incorporating the Conditional Network, the cINN framework is able to bridge the gap between the simulated model behaviour and the real-world observations, allowing for more accurate and efficient model updating. The combined effect of the INN's invertibility and the Conditional Network's conditioning capabilities makes the cINN particularly powerful in scenarios where traditional Bayesian or optimisation-based methods might struggle with computational inefficiencies or require large amounts of data.

A more detailed illustration of how the conditional network is combined with the ACL in each basic layer is shown in Fig. 4. It is observed that two ACLs are combined along permutation directions, with each set of ACLs linked to a single layer of the conditional

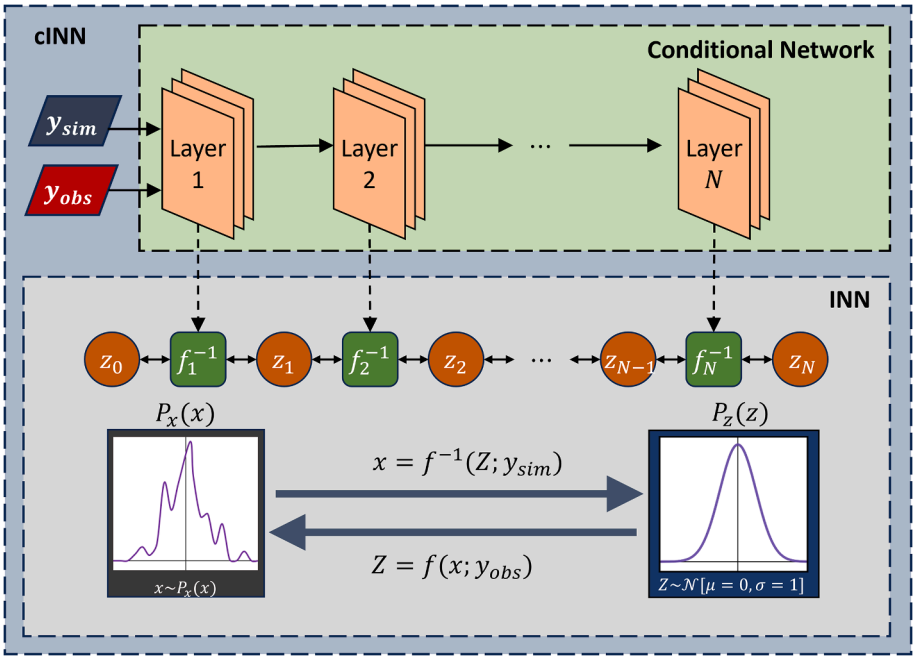


Fig. 3. The connection between the conditional network and the INN.

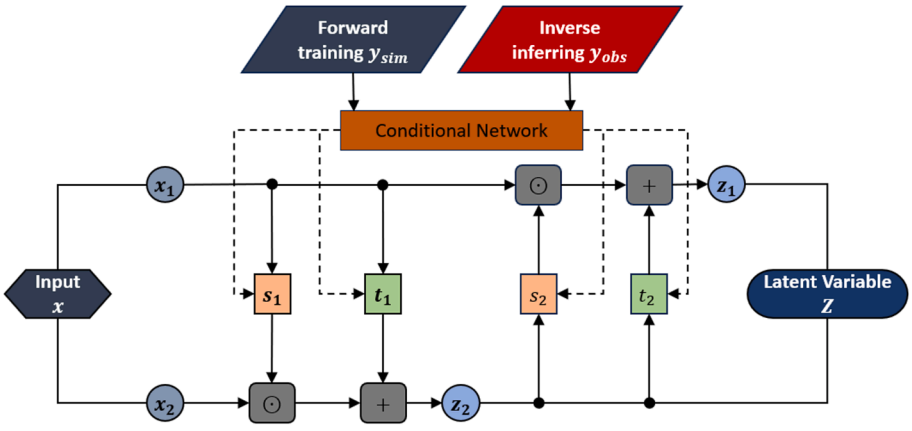


Fig. 4. The structure of cACL.

network. This basic unit is known as a conditional Affine Coupling Layer (cACL). Similar to the transformation of a single ACL explained in Sec 3.2.1, the forward training (with $\boldsymbol{y}_{sim}$) and inverse inferring (with $\boldsymbol{y}_{obs}$) of the cACL unit can be expressed as the following equations:

$$\begin{aligned} z_1 &= x_1 \odot \exp(s_2(z_2, \boldsymbol{y}_{sim})) + t_2(z_2, \boldsymbol{y}_{sim}) \\ z_2 &= x_2 \odot \exp(s_1(x_1, \boldsymbol{y}_{sim})) + t_1(x_1, \boldsymbol{y}_{sim}) \end{aligned} \tag{11}$$

$$\begin{aligned} x_1 &= (z_1 - t_2(z_2, \boldsymbol{y}_{obs})) \oslash \exp(s_2(z_2, \boldsymbol{y}_{obs})) \\ x_2 &= (z_2 - t_1(x_1, \boldsymbol{y}_{obs})) \oslash \exp(s_1(x_1, \boldsymbol{y}_{obs})) \end{aligned} \tag{12}$$

The cINN is constructed by stacking multiple N cACLs one after another, as illustrated in Fig. 3. The INN is parameterised by parameters ϕ. Let $\mathbb{S}_\varphi(\bullet)$ denotes the conditional network with network parameters φ, it can learn the most meaningful summary information ($\mathbb{S}_\varphi(\boldsymbol{y})$) directly from the raw simulation/observation data $\boldsymbol{y}$, rather than relying on limited hand-crafted summary statistics that may lose information. The parameters $[\phi, \varphi]$ represent the construction coefficients of the INN and the conditional network (e.g. connection weights, activation threshold, and scaling factors among layers), which are to be optimised during training. The objective of the training process is to find optimal values for parameters $[\phi, \varphi]$ so that the cINN is able to provide accurate predictions.

3.2.3. *Training the cINN*

In the process of training, the common but effective distribution matching training method is adopted. The estimated posterior distribution of the input parameter is computed as Eq. (13), according to the CoVR.

$$P(x|\boldsymbol{y}_{sim}) = P_Z(Z) \cdot \prod_{i=1}^{N} \left| \det \left(\frac{\partial f_i(Z_{i-1}, \mathbb{S}_\varphi(\boldsymbol{y}_{sim}))}{\partial Z_{i-1}} \right) \right| \tag{13}$$

where N represents the number of cACLs; f_i refers the i th cACL taking both the latent variable, Z_{i-1} and the summarised training output data, $\mathbb{S}_\varphi(\boldsymbol{y}_{sim})$, as input. Therefore, the objective of the training is to maximise the matching between the original training sample distribution $P_x(x_{train})$ and the estimated posterior distribution $P(x|\boldsymbol{y}_{sim})$, which can be quantified by minimising the Kullback-Leibler (KL) divergence between the two distributions. The loss function is then defined below.

$$L = KL[P_x(x_{train}) \| P(x|\boldsymbol{y}_{sim})] \tag{14}$$

Subsequently, the objective function of this optimisation problem is written in Eq. (15). The optimised construction coefficients of the conditional network and the INN are obtained by solving the optimisation equation below, which refers to minimise the expectation of the KL-divergence between the estimated and the training distribution of the input parameters in the field of all conditional data.

$$\widehat{\phi}, \widehat{\varphi} = \underset{\phi,\varphi}{\operatorname{argmin}} \mathbb{E}_{P(\boldsymbol{y}_{sim})} \{ \mathbb{KL}[P_x(x_{train}) \| P(x|\boldsymbol{y}_{sim})] \} \tag{15}$$

The Eq. (15) can be further transformed into Eq. (16) regarding the calculation of KL-divergence.

$$\widehat{\phi}, \widehat{\varphi} = \underset{\phi,\varphi}{\operatorname{argmin}} \mathbb{E}_{P(\boldsymbol{y}_{sim})} \{ \mathbb{E}_{P_x(x_{train})} [\log P_x(x_{train}) - \log P(x|\boldsymbol{y}_{sim})] \} \tag{16}$$

Because the original training sample distribution $P_x(x_{train})$ is generally independent of the estimated posterior distribution $P(x|\boldsymbol{y}_{sim})$, the $\mathbb{E}_{P_x(x_{train})}[\log P_x(x_{train})]$ remains constant. The Eq. (16) is then rewritten as Eq. (17).

$$\widehat{\phi},\widehat{\varphi}=\underset{\phi,\varphi}{\operatorname{argmax}}\mathbb{E}_{P(\boldsymbol{y}_{sim})}\left\{\mathbb{E}_{P_X(x_{train})}[\log P(x|\boldsymbol{y}_{sim})]\right\} \tag{17}$$

The $\log P(x|\boldsymbol{y}_{sim})$ can be computed through taking the logarithm of Eq. (13).

$$\widehat{\phi},\widehat{\varphi}=\underset{\phi,\varphi}{\operatorname{argmax}}\iint P(x_{train},\boldsymbol{y}_{sim})\cdot\log P(x|\boldsymbol{y}_{sim})dx_{train}d\boldsymbol{y}_{sim} \tag{18}$$

$$=\underset{\phi,\varphi}{\operatorname{argmax}}\iint P(x_{train},\boldsymbol{y}_{sim})\left(\log P_Z\left(Z=f_{\phi,\varphi}(x)\right)+\log\left|\det\boldsymbol{J}_{\phi,\varphi}\right|\right)dx_{train}d\boldsymbol{y}_{sim} \tag{19}$$

By adopting a standard Gaussian distribution as the latent distribution and a size of M training dataset $\{x^m,y^m\},m=1,2,\cdots,M$, the integral above can be computed as Eq. (20) below, where f_ϕ refers the cACLs with their construction coefficients ϕ,φ, and $\boldsymbol{J}_{\phi,\varphi}$ refers to the corresponding Jacobian matrix. The optimal values of the network's construction coefficients are finally obtained.

$$\widehat{\phi},\widehat{\varphi}=\underset{\phi,\varphi}{\operatorname{argmin}}\frac{1}{M}\sum_{m=1}^{M}\left(\frac{\left|f_{\phi,\varphi}\left(x_{train}^m;\mathbb{S}_\varphi(\boldsymbol{y}_{sim}^m)\right)\right|^2}{2}-\log\left|\det\boldsymbol{J}_{\phi,\varphi}^m\right|\right) \tag{20}$$

4. Stochastic damage identification with cINN-based model updating

The damage detection framework presented in this paper is developed using a multilevel stochastic model updating approach. As shown in Fig. 5, the framework consists of two rounds of model updating, the evaluation of the Probability of Damage (PoD), and a final step involving stochastic detection of structural damage. The process begins with initial model updating, where a numerical model is established based on prior knowledge of the original, undamaged structure. The initial numerical model normally differs from the physical system and hence requires to be updated by incorporating experimental measurements. This step ensures that the updated initial model accurately represents the original state of the structure and serves as a baseline for subsequent comparisons.

Following the initial model updating, the structure is placed under arbitrary in-service conditions. These conditions may include

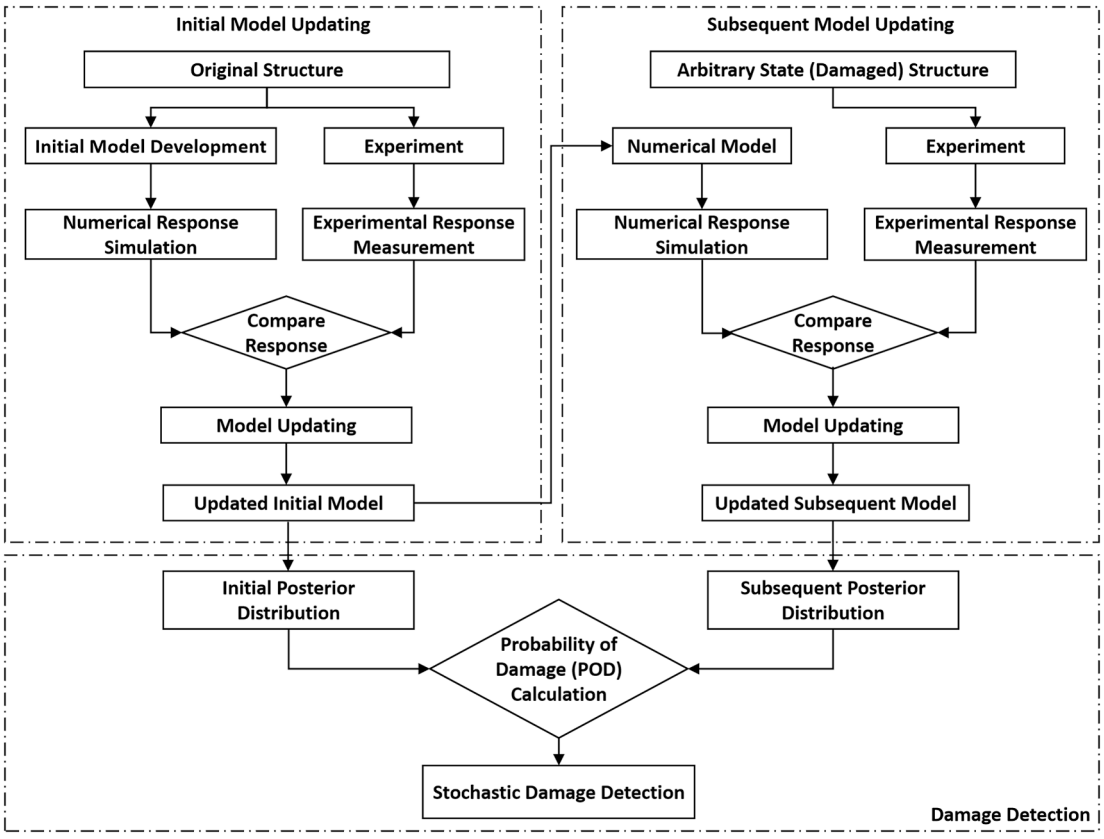


Fig. 5. Flowchart of the stochastic damage detection process.

loading scenarios, environmental effects, or possible degradation or damage to the structure. New experimental measurements are taken, reflecting the structure's current state under these conditions. These data are then used to further update the numerical model, resulting in a subsequent updated model. This step allows for capturing any changes in the structure's response that may have occurred due to damage or other factors. The updated model reflects the current state of the structure and provides the basis for detecting any significant deviations from the original condition.

To detect damage, the distribution of the structural parameters from both the initial and subsequent models is compared, and subsequently, the PoD can be evaluated. The PoD quantifies the likelihood that the structure has sustained damage based on the observed changes in its behaviour. This is a critical step, as it provides a probabilistic measure of the confidence in the damage detection process, accounting for uncertainties in both the measurements and the model predictions.

This paper defines PoD as the probability that a given structural parameter falls below a pre-set threshold, which indicates a degradation in its value. This threshold is determined based on the probabilistic distribution of the parameter in the undamaged (initial) state. The structural parameter, such as stiffness, is assumed to follow a specific probability distribution (typically a Gaussian distribution) with the mean and variance updated from the initial model updating process.

To explain this process, consider the example of a stiffness parameter in the undamaged condition. After the initial model updating, the stiffness is assumed to follow a Gaussian distribution $k\ N(4.0,0.3^2)$, as shown in Fig. 6. Note that, even in the undamaged case, due to the inherent uncertainties in both the model and the experimental observations, it is unrealistic to say the PoD is zero. Instead, a low but non-zero probability is associated with the undamaged condition. In this framework, we define a threshold probability p_{thre} to represent this "safe" state, and in this paper, we set this threshold probability to $p_{thre} = 5\%$. The damage threshold δ, corresponding to the threshold probability p_{thre}, can hence be determined as

$$\delta = \delta|_{p_{thre}=0.05} = \mu_{ud} - 1.645\sigma_{ud} \tag{21}$$

For the stiffness parameter example, as shown in Fig. 6, the 5 % threshold corresponds to a stiffness value of 3.5065, as calculated from the Gaussian distribution $k\ N(4.0,0.3^2)$. In Fig. 6, the red vertical line indicates the location of this threshold. The area under the curve to the left of this threshold represents the 5 % chance of the parameter being below this value in the undamaged state.

When evaluating the structure's condition after it has been subjected to arbitrary in-service conditions, the PoD is calculated by comparing the updated posterior distribution of the parameter with this predefined threshold. If the updated parameter distribution shifts significantly, such that a higher portion of the probability mass lies below the threshold, the PoD increases, indicating a greater likelihood of damage. Conversely, if the updated distribution remains close to the initial undamaged distribution, the PoD remains low, suggesting that the structure is likely still in a safe condition.

The PoD provides a probabilistic measure of the likelihood that a structural parameter has degraded beyond a critical threshold, taking into account the uncertainties in both the model and the observations. By defining this threshold based on the undamaged condition and calculating the shift in the posterior distribution, the PoD enables a quantitative assessment of structural health and the risk of damage.

5. Simulated case study: A 3-DOF mass-spring example

5.1. Problem description

In this study, a classic 3-degree-of-freedom spring-mass system (shown in Fig. 7) is introduced as the case study for stochastic

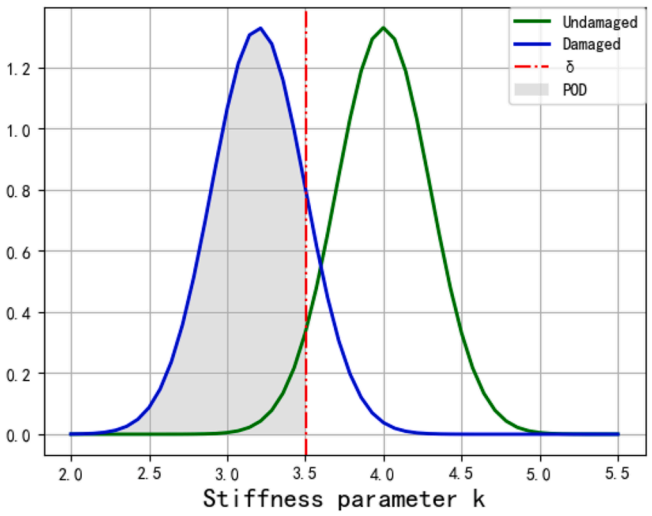


Fig. 6. Schematic diagram of the probability of damage (PoD).

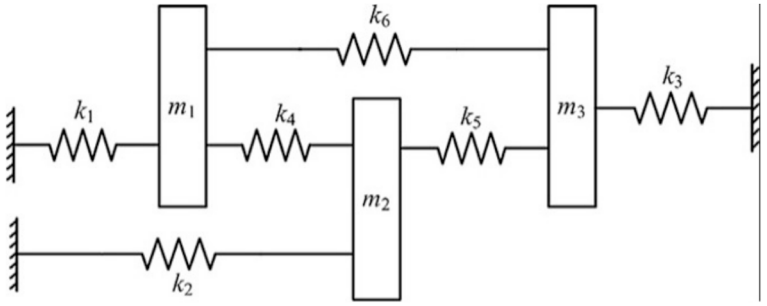


Fig. 7. 3-DOF spring-mass system.

damage detection by utilising the cINN-based model updating method introduced before. The corresponding mass and stiffness matrices are expressed in Eq. (22). All the springs in the system are assumed to follow Hooks's law and the linear elasticity assumption. The three stiffness coefficients k_4, k_5, and k_6 and the three masses m_1, m_2 and m_3 are set to be constant variables, where $k_4 = k_5 = k_6 = 5.0N/m$, $m_1 = 0.7kg$, $m_2 = 0.5kg$, and $m_3 = 0.3kg$. The other three stiffness coefficients k_1, k_2, and k_3 are random variables with uncertainty to be calibrated in this example. The three natural frequencies f_1, f_2, and f_3 are the quantity of interest whose uncertainties are driven by the uncertain parameters.

$$\mathbf{M} = diag(m_1, m_2, m_3)$$

$$K = \begin{bmatrix} k_1 + k_4 + k_6 & -k_4 & -k_6 \\ -k_4 & k_2 + k_4 + k_5 & -k_5 \\ -k_6 & -k_5 & k_3 + k_5 + k_6 \end{bmatrix} \tag{22}$$

Three scenarios of damage are simulated by various degrees of stiffness reduction at different locations among $k_1 - k_3$. In the first scenario, damage is located on k_1 with a stiffness reduction of 10%; damages are located on both k_2 and k_3 with a stiffness reduction of 10% and 20% separately in Scenario 2; all of the three stiffness are damaged in the 3rd scenario with a stiffness reduction of 10%, 10%, and 20% on k_1, k_2, and k_3, respectively, as illustrated in Table. 1. The objective of this example is to assess whether the cINN-based approach can accurately detect and locate damage in the stiffness parameters through the model updating process, and if it can correctly predict the probability of damage (PoD) in the different damage scenarios. The performance of the cINN-based approach is compared with the sensitivity approach and the Bayesian approach at the end of this section.

5.2. Stochastic damage detection with cINN

In the stochastic damage detection with cINN-based multilevel model updating, the parameters to be calibrated are the means ($\mu_1 - \mu_3$) and standard deviations ($\sigma_1 - \sigma_3$) of the three stiffness parameters ($k_1 - k_3$). The target distributions of the stiffness parameters are Gaussian distributions with different means and variance values, the prior distribution of the parameters is assumed to be a uniform distribution, as illustrated in Table. 2. The model updating can provide the posterior distribution of the stiffness parameters, and the mean and standard deviation values can therefore be evaluated through the posterior samples. By comparing the posterior distributions of the stiffness parameters under different statuses of the system (intact and arbitrarily damaged), damage identification and localisation can be achieved, and the PoD can then be calculated according to the process introduced in the section above.

In the training phase, the conditional network and the INN were trained jointly by adopting a maximum likelihood approach to calibrate the network's construction coefficient. The definition of loss function is denoted in Section 3.2.2. The 1D CNN was chosen to serve as the conditional network, and the INN consists of 4 cACLs. The training dataset is generated by sampling a size of 10,000 samples from the prior distribution and collecting the corresponding simulation outputs. After training, the networks are optimised automatically so that they can build a bijective mapping between the input and latent variables accurately. The coefficient of determination (R^2) is employed to validate the accuracy of the training process. The training accuracy validation results are presented in Fig. 8, showing scatter plots of the trained cINN's accuracy in estimating stiffness parameters. Each point in these plots represents a single sample from the training dataset, with the horizontal axis ("Ground truth") indicating the true input parameter values used to generate the training data, and the vertical axis ("Estimated") showing the corresponding predicted values by the trained cINN. The

Table 1
Three scenarios of damage.

Scenario	Position	Severity
1	k_1	−10%
2	k_2, k_3	−10%, −20%
3	k_1, k_2, k_3	−10%, −10%, −20%

Table. 2
Uncertain characteristics of the 3-DOF system for cINN-based stochastic damage detection.

Parameters	Prior distribution	Target distribution
k_1	k_1 $U(3.0, 7.0)$	k_1 $\mathcal{N}(\mu_1 = 4.0, \sigma_1 = 0.3)$
k_2	k_2 $U(3.0, 7.0)$	k_2 $\mathcal{N}(\mu_2 = 5.0, \sigma_2 = 0.1)$
k_3	k_3 $U(3.0, 7.0)$	k_3 $\mathcal{N}(\mu_3 = 6.0, \sigma_3 = 0.2)$
$k_4 - k_6 m_1 - m_3$	Constant variables do not need to be updated. $k_4 = k_5 = k_6 = 5N/m, m_1 = 0.7kg, m_2 = 0.5kg, m_3 = 0.3kg$	

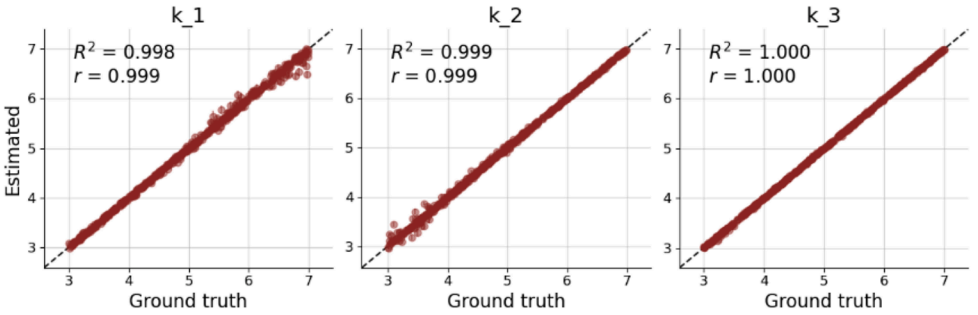


Fig. 8. Training accuracy validation.

close alignment of points along the diagonal reference line demonstrates that the cINN model has accurately captured the input–output relationship during training.

In the inference phase, the well-trained cINN can operate inversely to infer the distribution of input parameters by sampling randomly from the latent distribution with given observation data as the condition. Here, the observation data is synthetically generated using 50 random samples drawn from the target distribution of input parameters, as specified in Table 2. These 50 input parameter samples are then substituted into the analytical model, producing 50 corresponding samples of the natural frequencies. The histogram and fitted PDFs of these natural frequency samples are illustrated in Fig. 9.

There are 50 synthetic measurement frequency samples utilised in the model updating process, implying 500 samples of the stiffness parameters are obtained via the inverse application of the cINN architecture, where 10 samples are generated inversely based on a single measurement sample. The choice of generating 10 samples per measurement sample is case-dependent and lacks a standardised guideline. Generally, a larger number of generated samples is recommended when fewer observations are available, while fewer samples per observation are sufficient when numerous observations are present.

The PDFs of the stiffness parameter samples (denoted as "Posterior") are estimated and illustrated in Fig. 10, alongside the target distributions. It is observed that the obtained posterior PDFs closely match the target distributions, indicating an effective calibrating process by the cINN-based model updating. Table 3 presents the hyperparameters (mean and standard deviation) of the stiffness parameter samples. It is noted that the error between the updated hyperparameters and their targets is minimal.

As previously mentioned, 3 different damage scenarios are simulated as different stiffness reductions at different places and the

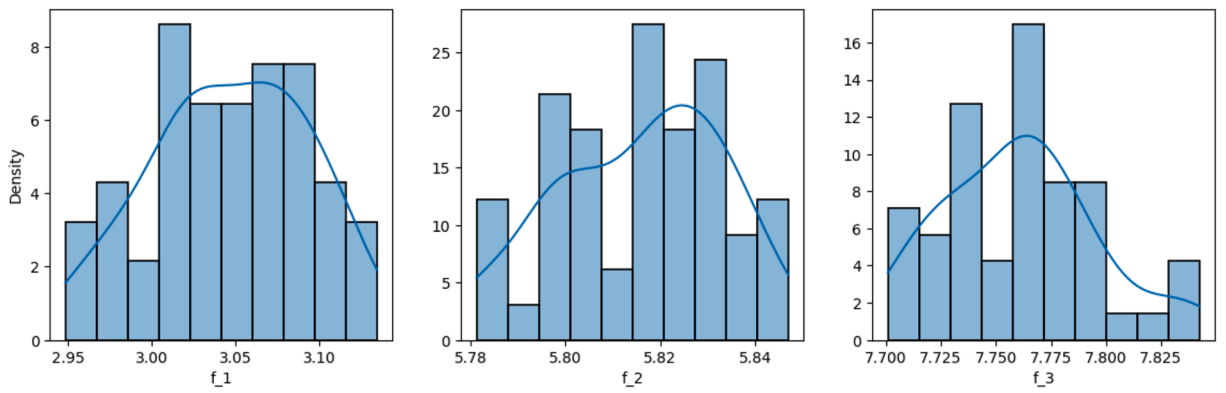


Fig. 9. Histogram for the synthetic observation data (undamaged).

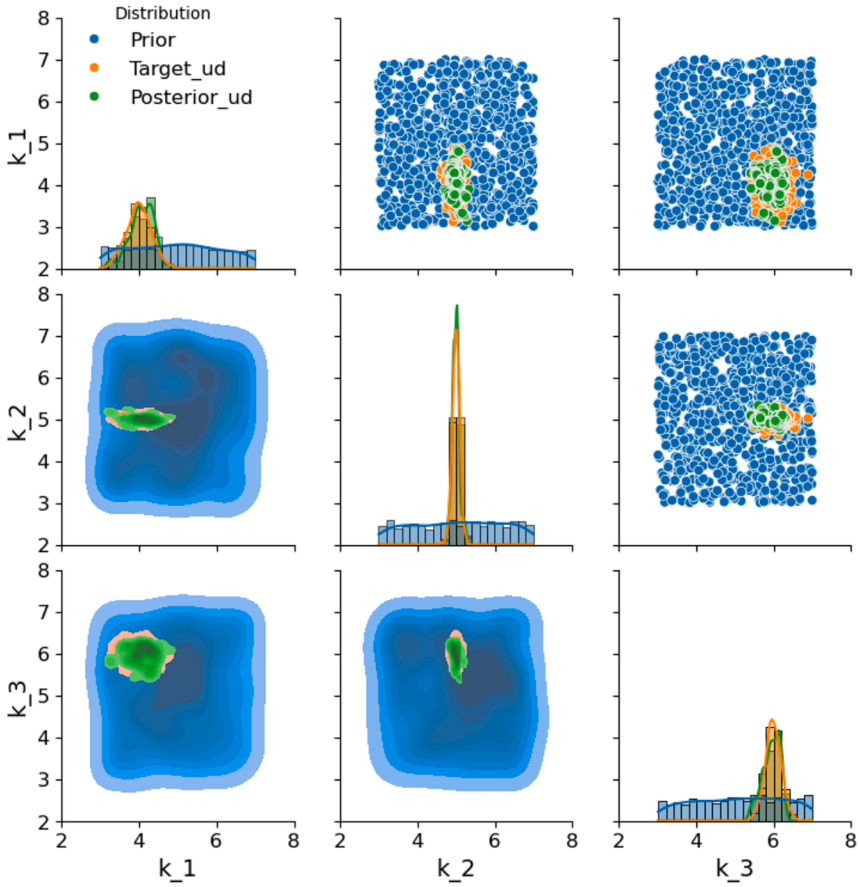


Fig. 10. Posterior histogram and distributions of the stiffnesses compared with the prior and target distributions.

Table 3
Mean and standard deviation of the stiffness parameters from the updated and target data.

Parameters	Target value		Updated result	
	Mean	Std	Mean	Std
k_1	$\mu_1 = 4.0$	$\sigma_1 = 0.3$	$\mu_1 = 4.083$	$\sigma_1 = 0.314$
k_2	$\mu_2 = 5.0$	$\sigma_2 = 0.1$	$\mu_2 = 5.008$	$\sigma_2 = 0.097$
k_3	$\mu_3 = 6.0$	$\sigma_3 = 0.2$	$\mu_3 = 5.987$	$\sigma_3 = 0.188$

corresponding natural frequencies under different damage scenarios are considered as the observation data. The histograms for the simulated natural frequencies under three damage scenarios are demonstrated in Fig. 11 below. The cINN-based model updating is carried out on each damage scenario and the model updating process is the same as described above. The obtained distributions of the stiffness parameters are shown in Fig. 12 with their mean and standard derivations listed in Table. 4. Similar to the model updating results of the undamaged case, it is observed that all three damaged scenarios have satisfied updating outcomes since the posterior PDFs fit with the target PDF well and their hyperparameters are close to the target values.

The posterior distributions of the stiffness parameters under different damage scenarios are presented in Fig. 13. The PoD can therefore be determined by calculating the integral of the Probability Density Function (PDF) for stiffness values below the pre-defined damage threshold δ. The threshold δ is defined following Eq. (21) with $\delta = \mu_{ud} - 1.645\sigma_{ud}$.

In the Damage Scenario 1 (Fig. 13(a)), where only k_1 is damaged (as specified in Table 1), the predicted PDFs accurately reflect this scenario, showing a significant difference between the undamaged PDF and the damaged PDF for k_1. In Damage Scenario 2(Fig. 13(b)), the damages in both k_2 and k_3 are detected, as evidenced by the clear shifts in their damaged PDFs compared to the undamaged case. Similarly, in Damage Scenario 3 (Fig. 13(c)), the damages in k_1, k_2 and k_3 are all detected, with their respective damaged PDFs distinctly shifted from the undamaged PDFs. The predicted PoDs are calculated according to the predicted PDFs and the pre-set damage threshold and are listed in Table 4. It is observed that the predicted PoDs are similar to the target PoDs, implying that the proposed approach is capable of accurately predicting the actual damage situation.

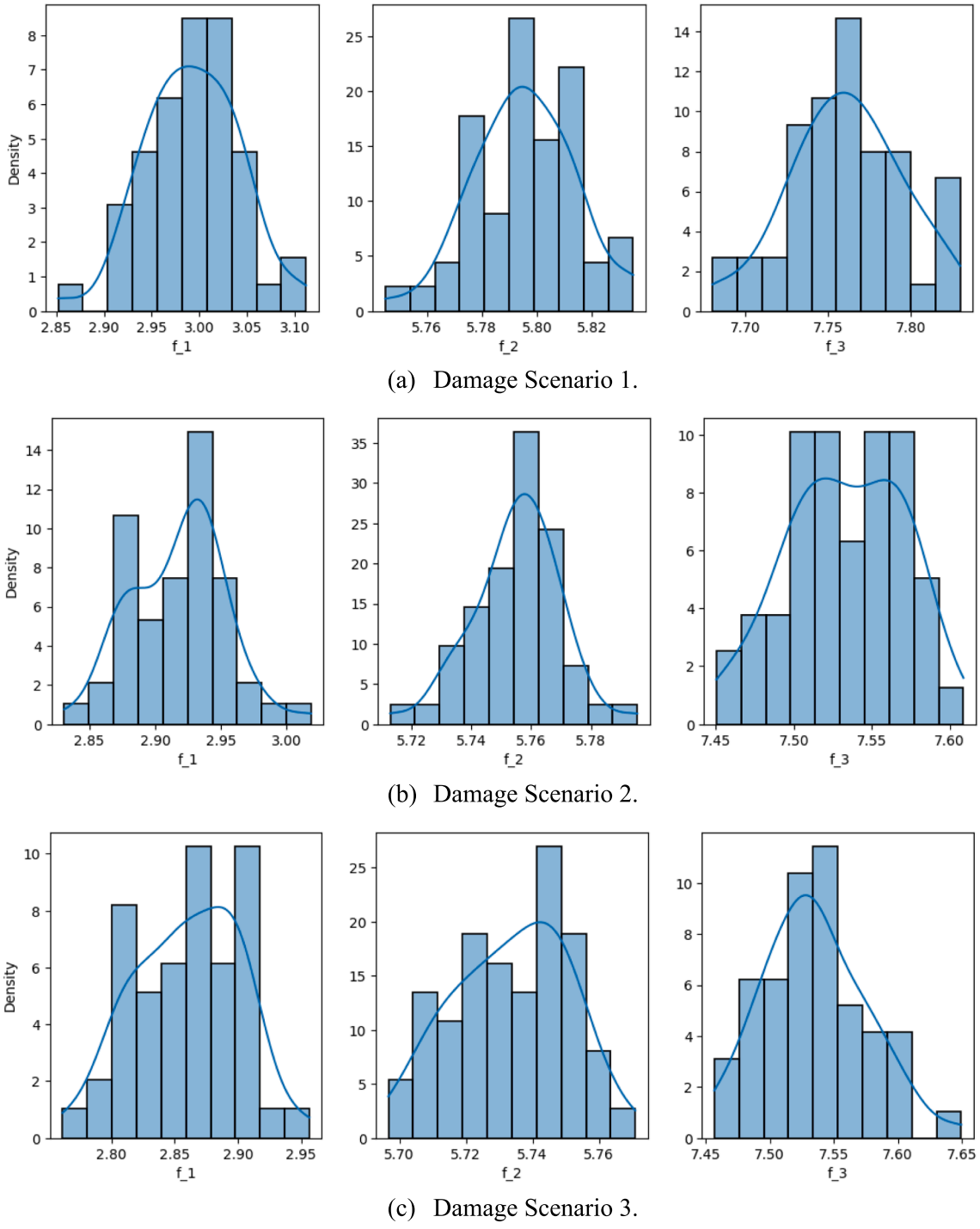
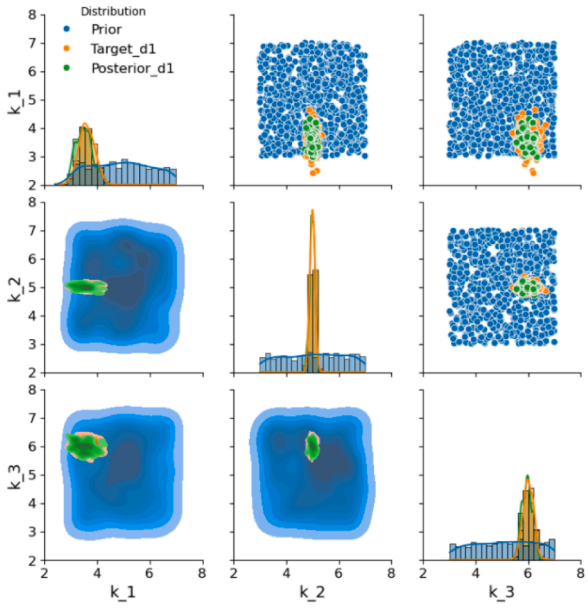


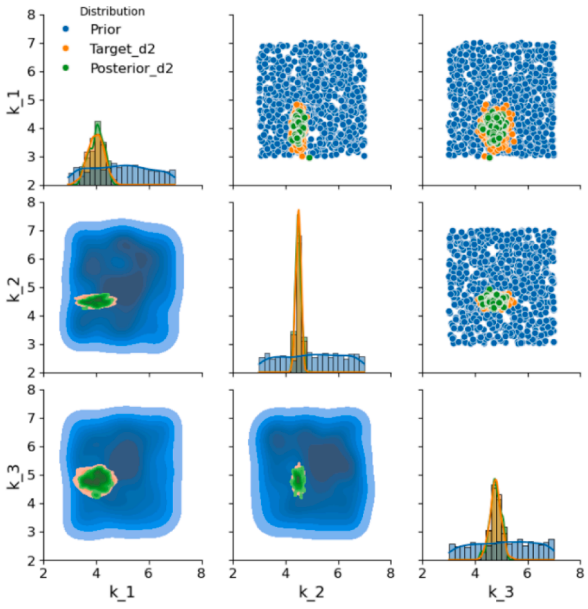
Fig. 11. Histogram for the synthetic observation data (damaged).

5.3. Comparison with sensitivity-based approach and Bayesian approach

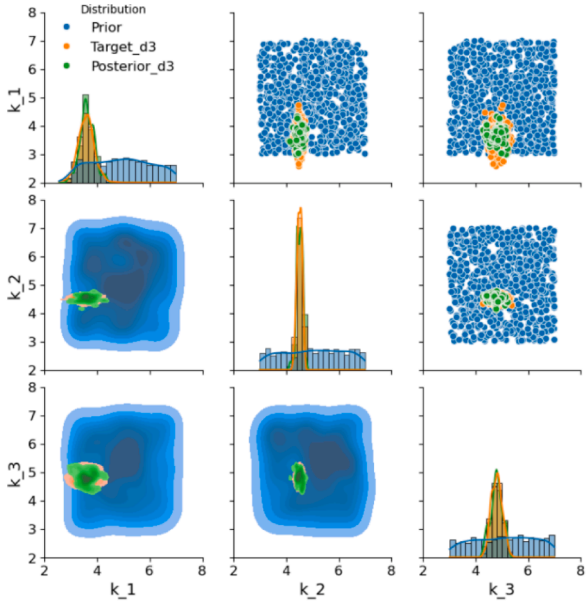
The model updating results under undamaged scenarios derived from the sensitivity-based approach and Bayesian approach are provided for comparison with the proposed cINN updating. A total of 500 sets of synthetic observation data are adopted for the posterior estimation in the three methods. In the sensitivity-based approach, 500 sets of optimised input parameters are obtained based on the observation data for posterior approximation. In the Bayesian approach, the transitional Markov chain Monte Carlo (TMCMC) is adopted as the sampling method with a number of samples for 1000, and the likelihood function is defined based on the Bhattacharyya



(a) Damage Scenario 1.



(b) Damage Scenario 2.



(c) Damage Scenario 3.

Fig. 12. Prior, posterior, and target distributions of stiffness parameters $\boldsymbol{k}_1, \boldsymbol{k}_2$, and $\boldsymbol{k}_3$ under 3 damage scenarios.

distance.

The model updating results for all three methods are compared in Fig. 14 and Table. 5. All three methods can provide reliable updating results in this case with an acceptable error for both mean and standard deviation values. However, there are some differences among these methods. The sensitivity-based approach contains a series of deterministic processes to produce a dataset, where the mean and standard deviation values are estimated. The Bayesian approach can provide the marginal of the hyperparameters that contain more comprehensive information about the posterior. The cINN-based approach is able to generate a posterior distribution based on each set of observation data, and the total posterior distribution is obtained according to all the observation data.

Table 4
Target and predicted mean and standard deviation of the stiffness parameters, and PoD under 3 damage scenarios.

Scenarios	Parameters	Target value		Target PoD	Updated result		Predicted PoD
		Mean	Std		Mean	Std	
1	k_1	$\mu_1 = 3.6$	$\sigma_1 = 0.3$	37.76 %	$\mu_1 = 3.606$	$\sigma_1 = 0.327$	45.6%
2	k_2	$\mu_2 = 4.5$	$\sigma_2 = 0.1$	99.96 %	$\mu_2 = 4.511$	$\sigma_2 = 0.091$	100%
	k_3	$\mu_3 = 4.8$	$\sigma_3 = 0.2$	99.99 %	$\mu_3 = 4.794$	$\sigma_3 = 0.202$	100%
3	k_1	$\mu_1 = 3.6$	$\sigma_1 = 0.3$	37.76 %	$\mu_1 = 3.581$	$\sigma_1 = 0.275$	54.6%
	k_2	$\mu_2 = 4.5$	$\sigma_2 = 0.1$	99.96 %	$\mu_2 = 4.520$	$\sigma_2 = 0.090$	100%
	k_3	$\mu_3 = 4.8$	$\sigma_3 = 0.2$	99.99 %	$\mu_3 = 4.826$	$\sigma_3 = 0.171$	100%

The computational time required for the model updating process using different methods is summarised in Table 6. The sensitivity-based approach converges rapidly due to its deterministic nature. In contrast, the Bayesian approach takes significantly longer to reach convergence because of its reliance on sampling methods. The cINN-based approach requires more time for training compared to the sensitivity-based method, but it is considerably faster than the Bayesian sampling approach. Once trained, the cINN model can perform posterior estimation quickly in the inference phase. Overall, the cINN-based approach demonstrates a well-balanced performance, providing an effective trade-off between computational efficiency and prediction accuracy.

6. Experimental case study: A 3-DOF experimental rig

6.1. Problem description

This section uses the same 3-DOF experimental rig setup designed and utilised as a bench test for the model updating methods described in Ref [38], as shown in Fig. 15. In this setup, three masses are supported and connected to an anti-vibration table by three grounded plate-like springs, each consisting of two plates. The masses are interconnected by two coupling springs, each composed of two leaf springs joined by a rigid horizontal link. This link slides vertically to adjust the stiffness of the connection. The stiffnesses of the two coupling springs denoted as k_{12} and k_{23}, vary with the positions of the two horizontal linkages, i.e. p_{12} and p_{23}.

In the subsequent updating problem, the three messes $m_1, m_2,$ and m_3, along with the three ground springs k_{g1}, k_{g2}, and k_{g3}, are treated as known constants. The positions of the two rigid linkages, p_{12} and p_{23}, are considered as random variables following a pre-determined prior distribution. The nominal values and distribution characteristics of these variables are presented in Table. 7. Consequently, the stuffiness of the two coupling springs, k_{12} and k_{23}, are also random variables governed by the distribution features of p_{12} and p_{23}. It is essential to identify the functional relationship between these positional variables (p_{12} and p_{23}) and the corresponding stiffness variables (k_{12} and k_{23}). This relationship will be explored in the next section through a sensitivity analysis.

Once the masses and stuffiness are determined, the mass and stiffness matrices can be constructed using Eq. (23). The eigenvalue equation is then solved, allowing the natural frequencies of the system to be analytically obtained. These three natural frequencies serve as the output quantities of interest in the subsequent model updating and damage identification processes.

$$M = diag(m_1, m_2, m_3)$$
$$K = \begin{bmatrix} 2k_{g1} + k_{12} & -k_{12} & 0 \\ -k_{12} & 2k_{g2} + k_{12} + k_{23} & -k_{23} \\ 0 & -k_{23} & 2k_{g3} + k_{23} \end{bmatrix} \tag{23}$$

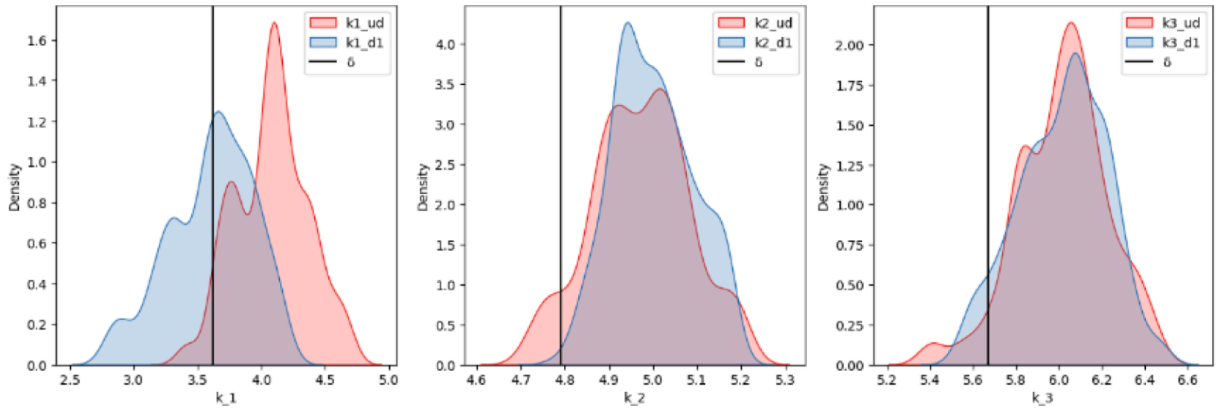
As depicted in Fig. 15(b), the position variables, p_{12} and p_{23}, follow a uniform distribution ranging from 0.02 m to 0.07 m. Initially, these positions were set at six evenly spaced intervals, resulting in 36 possible configurations (6-by-6). Subsequently, five additional intermediate positions were identified, creating 25 more configurations (5-by-5). This setup yielded a total of 61 configurations, and corresponding modal tests were conducted for each, resulting in 61 sets of experimental natural frequencies.

The observation data from these tests were categorised into two groups: 'undamaged' and 'damaged'. Configurations where either rigid linkage was in the top position (i.e., $p_{12} = 0.02m$ or $p_{23} = 0.02m$), indicating low stiffness in the coupling spring, were classified as 'damaged'. All other configurations were considered 'undamaged'. As a result, 50 sets of 'undamaged' data and 11 sets of 'damaged' data were obtained. Fig. 16 gives a clear view of how the 61 groups of observation data are measured and labelled.

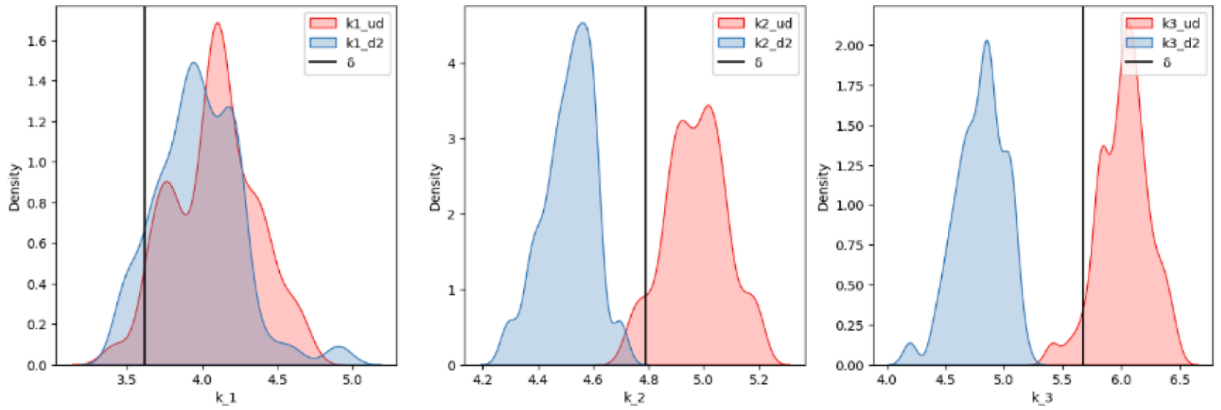
The goal of this experimental case study is to determine whether the cINN-based model updating approach can accurately detect and locate damage by distinguishing between these 'damaged' and 'undamaged' cases. The model updating results obtained by using the sensitivity method, Bayesian method, and the cINN-based method with 61 groups of experimental measurements as targets are compared at the end of this section.

6.2. Global sensitivity analysis

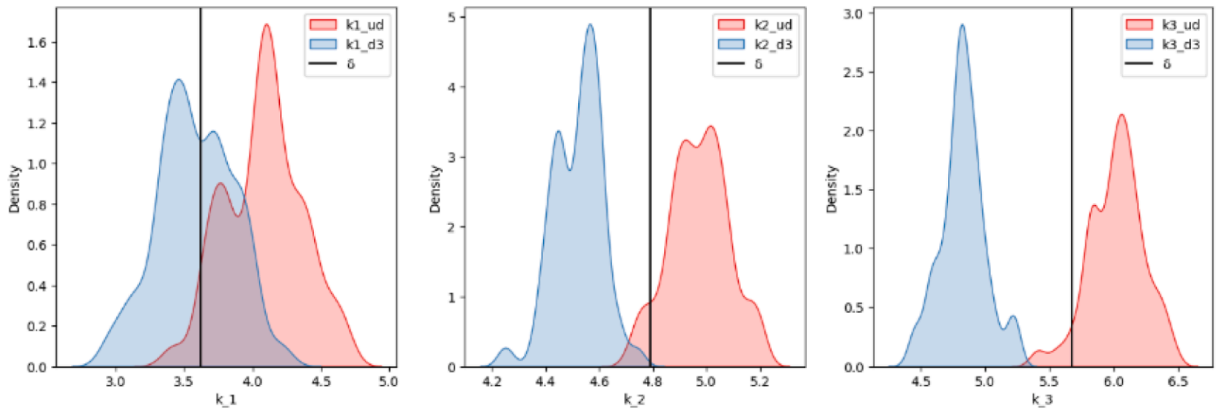
A global sensitivity analysis (GSA) was performed to identify the input parameters with the greatest aleatory variability, using Sobol total-order sensitivity indices. This approach decomposes the contribution of each input to the variance of the outputs. The results, illustrated in the heatmap in Fig. 17, show that the first natural frequency (f_1) is influenced by both position variables p_{12} and



(a) Damage Scenario 1.



(b) Damage Scenario 2.



(c) Damage Scenario 3.

Fig. 13. PoD evaluation based on updated PDFs of the stiffness from both undamaged case and damaged case (in the legend "k1-3_ud" indicates the undamaged case; "k1-3_d1-3" indicates the damaged case in different scenarios).

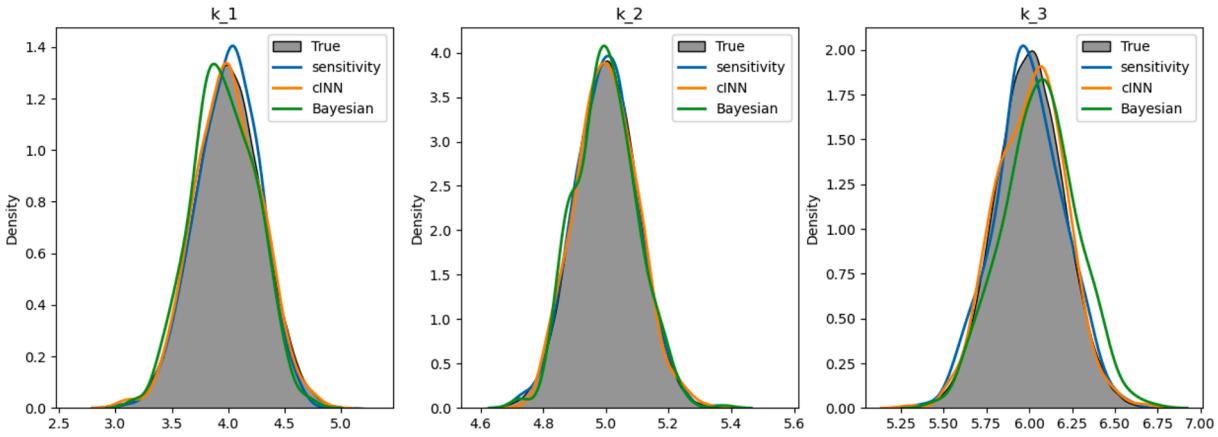


Fig. 14. Comparison for accuracy of model updating results from sensitivity, Bayesian, and cINN-based approaches.

Table 5
Means and standard deviations derived by sensitivity, Bayesian, and cINN-based approach.

Parameters			Updated result					
	Target value		Sensitivity-based		Bayesian[a]		cINN-based	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
k_1	$\mu_1 = 4.0$	$\sigma_1 = 0.3$	$\mu_1 = 3.998$	$\sigma_1 = 0.274$	$\mu_1 = 3.970$	$\sigma_1 = 0.279$	$\mu_1 = 4.083$	$\sigma_1 = 0.314$
k_2	$\mu_2 = 5.0$	$\sigma_2 = 0.1$	$\mu_2 = 4.999$	$\sigma_2 = 0.099$	$\mu_2 = 4.998$	$\sigma_2 = 0.101$	$\mu_2 = 5.008$	$\sigma_2 = 0.097$
k_3	$\mu_3 = 6.0$	$\sigma_3 = 0.2$	$\mu_3 = 5.998$	$\sigma_3 = 0.204$	$\mu_3 = 6.042$	$\sigma_3 = 0.215$	$\mu_3 = 5.987$	$\sigma_3 = 0.188$

[a] Results extracted from the Most Probable Points (MPPs) of the Bayesian posterior distributions.

Table 6
Comparison of time efficiency of sensitivity-based, Bayesian, and cINN-based approaches.

Approaches	Time-consumed[a]
Sensitivity-based	1m35s
Bayesian	53m22s
cINN-based	Train: 7m13s
	Inference: 12 s

[a] Computation conducted on a laptop equipped with an Intel(R) Core(TM) Ultra 135U processor.

p_{23}, with p_{23} being the more dominant factor.

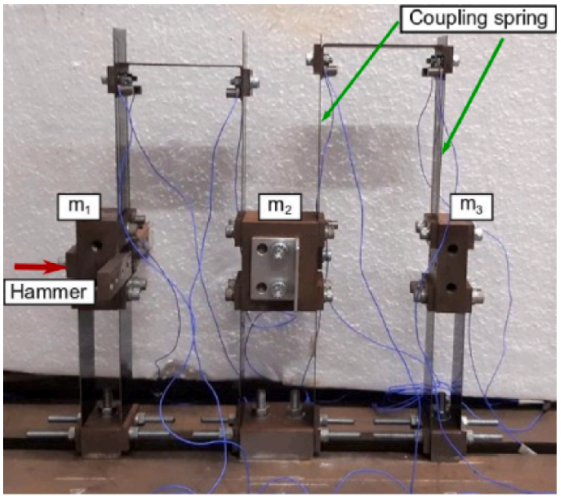
In contrast, the second and third natural frequencies (f_2 and f_3) are primarily influenced by a single parameter. Specifically, f_2 is driven predominantly by p_{12}, while f_3 is largely affected by p_{23}. The remaining parameters, including the ground spring constants (k_{g1}, k_{g2}, and k_{g3}) and mass variables ($m_1, m_2,$ and m_3), show minimal influence on the output variances, highlighting their lesser significance in this context.

Since the two eigenvalues, λ_2 and λ_3 are separately dependent on p_{12} and p_{23}, the values of coupling stiffness, k_{12} and k_{23}, may be determined by solving the system of characteristic equations in Eq. (24).

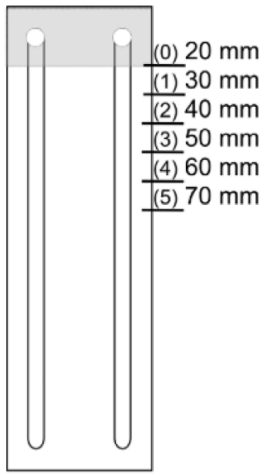
$$\begin{cases} |\mathbf{K} - \lambda_2 \mathbf{M}| = 0 \\ |\mathbf{K} - \lambda_3 \mathbf{M}| = 0 \end{cases} \tag{24}$$

where $\lambda_i = \left(2\pi \cdot f_i\right)^2$ and $f_i, i = 2, 3$ are the experimental data of the second and third frequencies. Thus, a correspondence between p_{12}, p_{23} and k_{12}, k_{23} are obtained as the blue dots in the figure below. Based on the trends in the scatter distribution, an exponential function was selected as the fitting function, expressed as Eq. (25). Fitted curves are expressed in Eq. (26) and illustrated in Fig. 18 in the red line.

$$k = A \cdot e^{B \cdot p} + C \tag{25}$$



(a) 3-DOF spring-mass system.



(b) Location of the rigid links (measured from the top of the leaf spring).

Fig. 15. 3-DOF experimental rig system [38].

Table 7
Uncertain characteristics of the 3-DOF experimental rig system.

Parameters	Prior distribution	Target distribution
p_{12}	Uniform distribution:U [0.0, 0.12]	Uniform distribution: U [0.02, 0.07].
p_{23}	Uniform distribution:U [0.0, 0.12]	Uniform distribution: U [0.02, 0.07].
m_1 $-m_3 k_{g1}$ $-k_{g3}$	Constant variables do not need to be updated.	
	$m_1 = 1.52kg, m_2 = 2.30kg, m_3 = 1.08kg$, $k_{g1} = k_{g2} = k_{g3} = 2477N/m$.	

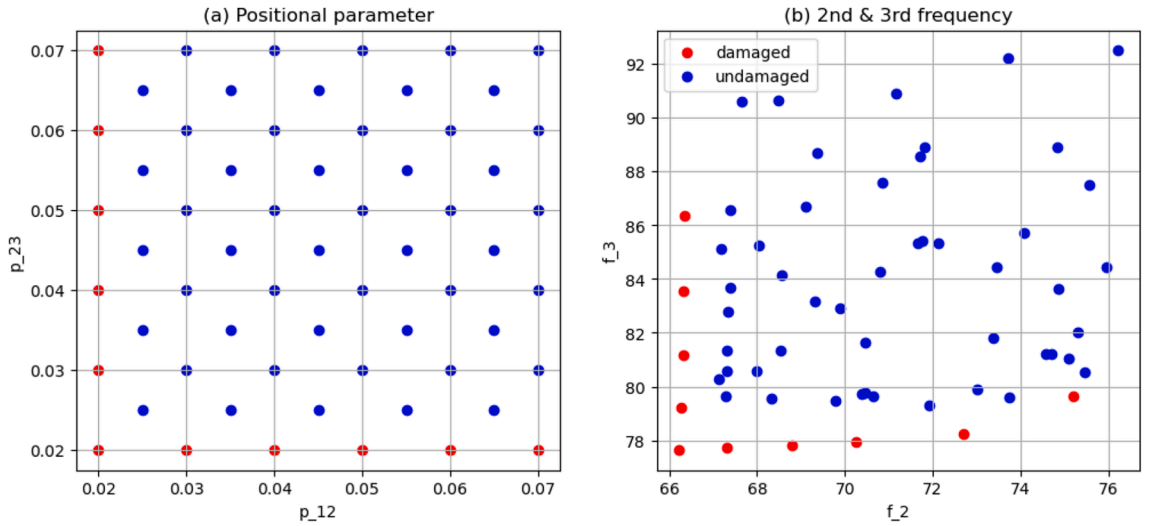


Fig. 16. Experimental dataset labelled with ‘undamaged’ and ‘damaged’.

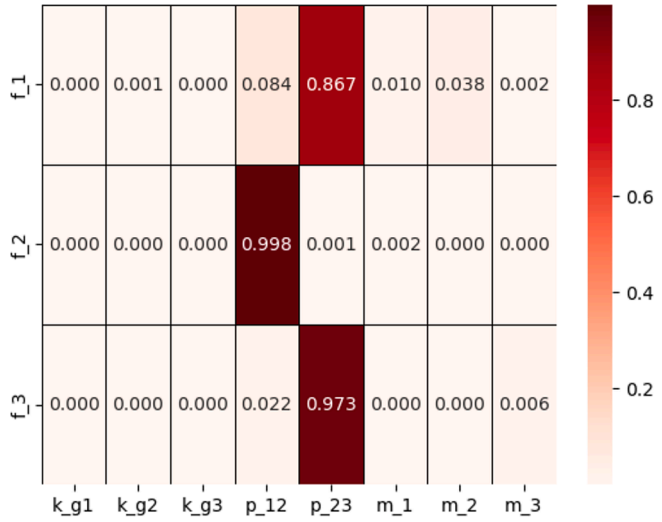


Fig. 17. Sobol total indices of the 8 parameters.

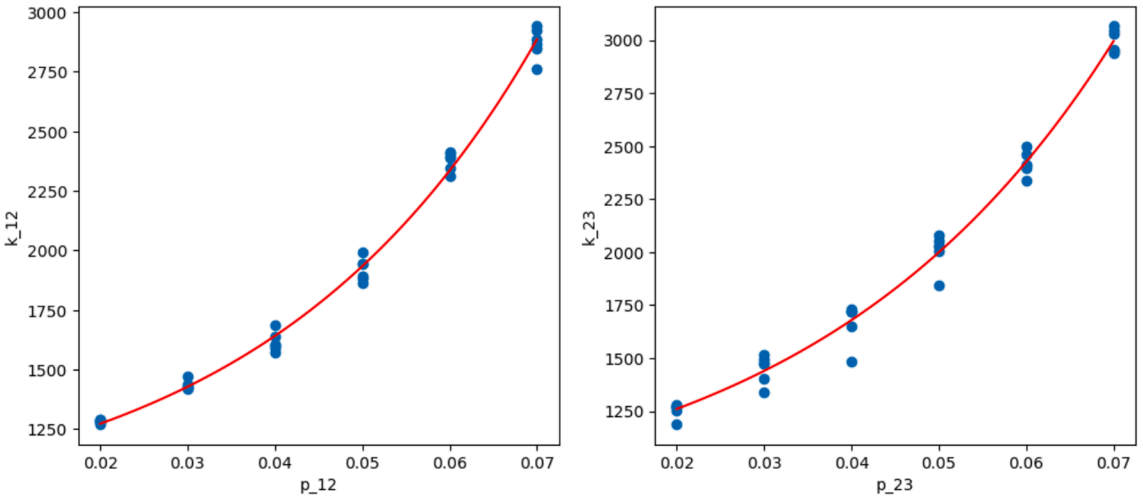


Fig. 18. Fitted expression of stiffness $\boldsymbol{k}_{12}$ and $\boldsymbol{k}_{23}$ with $\boldsymbol{p}_{12}$ and $\boldsymbol{p}_{23}$.

$$\begin{cases} k_{12} = 221.75 \cdot \exp(31.64 \cdot p_{12}) + 855.75 \\ k_{23} = 300.45 \cdot \exp(28.89 \cdot p_{23}) + 725.37 \end{cases} \tag{26}$$

6.3. *Stochastic damage detection with cINN*

Typically, traditional Bayesian model updating frameworks struggle when input parameters have varying levels of impact on the output features. This issue is particularly relevant in cases like the one revealed by the above sensitivity analysis, where the parameters p_{12} and p_{23} exhibit significantly different sensitivities across the three natural frequencies. To address this variability, parameters generally need to be updated in separate groups based on their impact, which can be a complex and challenging process. However, the cINN architecture offers an efficient solution to this challenge. The conditional network in the cINN is designed to provide learned data representations that are highly informative for the INN component. This capability allows the model to effectively capture and handle the varying sensitivities of different input parameters, thereby predicting the posterior distributions of the parameters with greater accuracy.

The training data is obtained by collecting the position parameters sampled randomly from the prior distribution as input and the corresponding natural frequencies generated from the simulator (numerical model) as conditional data. In the training phase, the conditional network is designed as a sequence of multiple Conv1D layers, and the INN consists of 4 cACLs. The validation results using the coefficient of determination (R^2) is shown in Fig. 19.

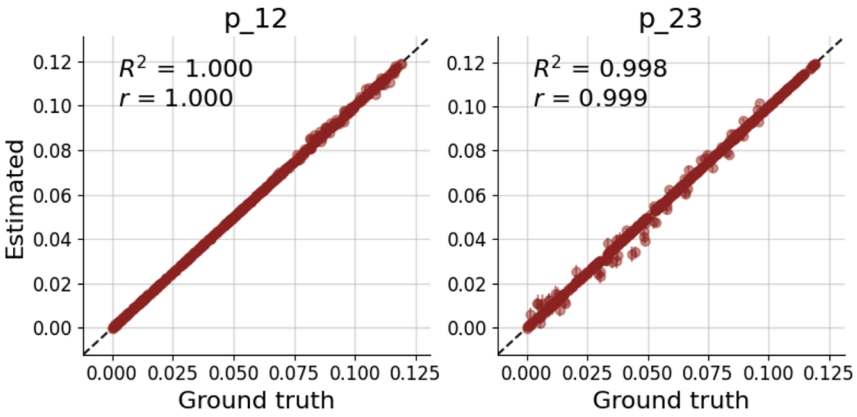


Fig. 19. Training accuracy validation of 3-DOF experimental rig.

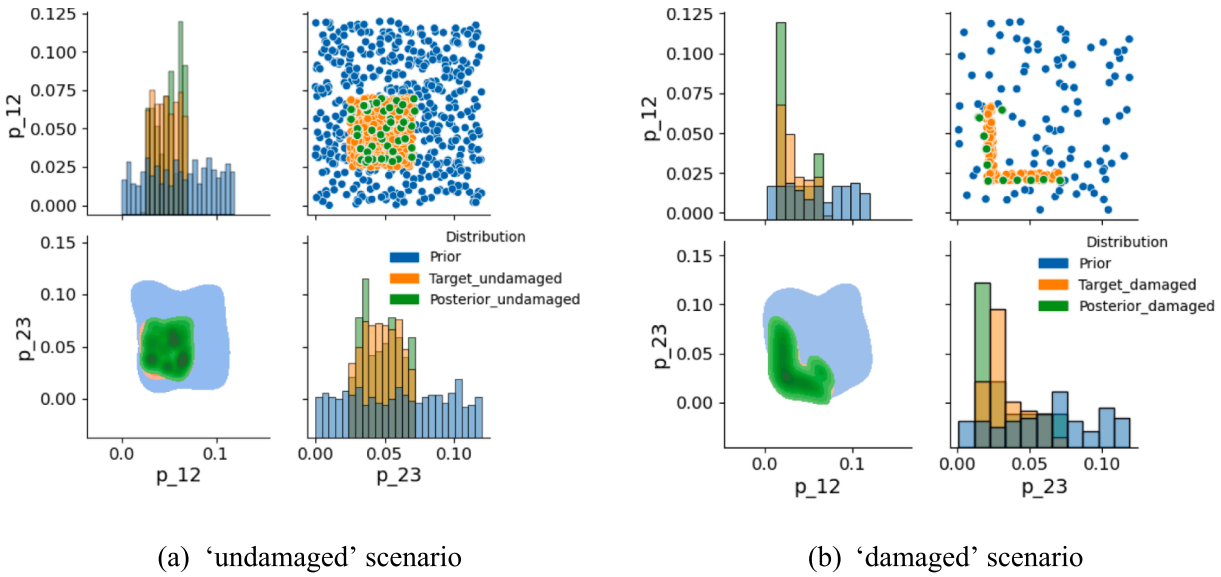


Fig. 20. Posterior distributions of $\boldsymbol{p}_{12}$ and $\boldsymbol{p}_{23}$ under 'undamaged' and 'damaged' scenario.

Model updating is initially conducted using the 'undamaged' data as conditional inputs for the well-trained cINN model to generate the posterior distributions of the position parameters p_{12} and p_{23}. For this process, 50 sets of 'undamaged' observation data are used, resulting in the generation of 500 data points, with each observation data point producing 10 inversely generated data points. As illustrated in Fig. 20(a), the estimated posterior distributions for p_{12} and p_{23} closely align with the target distributions. This alignment indicates that the cINN model accurately captures the characteristics of the system in its 'undamaged' state, thereby validating the effectiveness of the model calibration process.

Table 8
Target and predicted mean and standard deviation of $\boldsymbol{p}_{12}$ and $\boldsymbol{p}_{23}$, and PoD under'undamaged' and 'damaged' scenario.

Scenarios	Parameter	Target value		Target PoD	Updated result		Predicted PoD
		Mean	Std		Mean	Std	
'undamaged'	p_{12}	$\mu_{12} = 0.0475$	$\sigma_{12} = 0.0116$	5.0 %	$\mu_{12} = 0.0503$	$\sigma_{12} = 0.0137$	5.0%
	p_{23}	$\mu_{23} = 0.0475$	$\sigma_{23} = 0.0116$	5.0 %	$\mu_{23} = 0.0454$	$\sigma_{23} = 0.0147$	5.0%
'damaged'	p_{12}	$\mu_{12} = 0.0336$	$\sigma_{12} = 0.0177$	54.46 %	$\mu_{12} = 0.0334$	$\sigma_{12} = 0.0166$	56.36%
	p_{23}	$\mu_{23} = 0.0336$	$\sigma_{23} = 0.0177$	54.46 %	$\mu_{23} = 0.0337$	$\sigma_{23} = 0.0183$	45.46%

In the 'damaged' scenario, 11 sets of 'damaged' observation data serve as conditional inputs, leading to the generation of 110 data points for posterior estimation. Similar to the 'undamaged' case, each observation data point generates 10 data points inversely. As depicted in Fig. 20(b), the posterior distributions for both p_{12} and p_{23} peak distinctly at 0.02 m. This peak indicates a significant reduction in stiffness, which corresponds to a damaged state in the system. The 'L' shape on the scatter plot provides a clear view that either one of the two rigid links reaches the position $p = 0.02m$, the damage at the relative position is detected. The rest of the distribution appears relatively uniform, suggesting less variability in the other potential values. This uniformity further emphasises the anomaly detected in the system.

Table. 8 provides a detailed comparison of the updated mean and standard deviation values for p_{12} and p_{23} against their target values under both scenarios. The results show that the updated values are in close agreement with the target values, indicating that the cINN model has effectively captured the true underlying distribution of the position parameters. This close fit is critical, as it validates the accuracy of the cINN-based model updating process, confirming that the model can accurately predict the system's state based on the given observation data.

Furthermore, the Probability of Damage (PoD) is evaluated based on the posterior distributions of the stiffness parameters k_{12} and k_{23}, as illustrated in Fig. 21. The PoD calculation uses a threshold, $= \mu_{ud} - 1.645\sigma_{ud}$, where μ_{ud} and σ_{ud} are the mean and standard deviation of the 'undamaged' case. The threshold δ represents a critical value below which the system is considered damaged. In Fig. 21, the PDFs of k_{12} and k_{23} for both 'undamaged' and 'damaged' scenarios are shown. The PoD is determined by the area under the PDF curve to the left of the threshold δ. The observed significant differences in the PDFs between the 'undamaged' and 'damaged' cases, as well as the position of the threshold, clearly indicate the presence of damage. The PoD values provide a quantitative measure of the likelihood of damage, offering critical insights for structural health monitoring and maintenance decision-making.

6.4. *Comparison with sensitivity-based approach and Bayesian approach*

In this case, the sensitivity-based approach is carried out according to Ref [38] with the experimental dataset augmented by a further 500 natural frequencies sampled from the prior distribution. The Bayesian approach is carried out with TMCMC and Bhattacharyya distance-based likelihood function with all 61 sets of experimental measurement data as target. According to the GSA results, the 2nd and 3rd natural frequency is dependent on p_{12} and p_{23}, respectively, and the 1st natural frequency is dependent on both of the position parameters. Therefore, only the latter two order of natural frequencies are taken as the observation data while using sensitivity-based and Bayesian approach to perform model updating to ensure it is not ill-posed. For the cINN-based method, all three orders of natural frequencies are adopted as input of its conditional network directly. The results are compared in Fig. 22, Table. 9, and Table. 10, in the perspective of accuracy and efficiency.

Similar to the previous examples, all three methods yield relatively accurate model updating results. The sensitivity-based approach is the most time-efficient but does not provide a probabilistic representation of uncertainty. The cINN-based approach offers a strong balance between computational efficiency and UQ. Unlike the Bayesian method, which requires extensive sampling and is computationally expensive, the cINN-based approach directly obtains the posterior distribution through its invertible neural network structure, significantly reducing computation time while still capturing the probabilistic characteristics of the updated parameters. Once trained, the cINN model can infer posterior distributions quickly, making it a practical and scalable solution for stochastic model updating, particularly in real-time applications or cases with large-scale simulations.

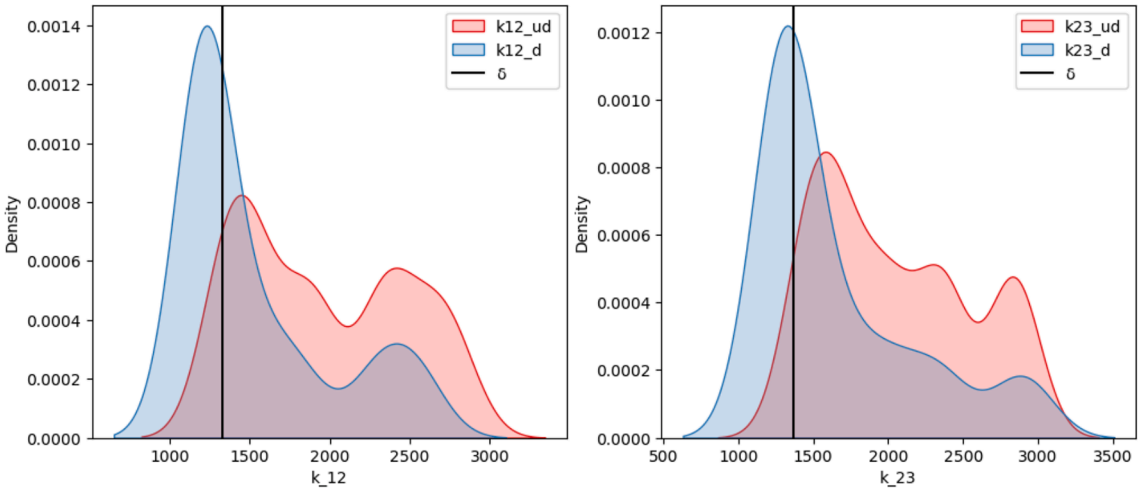


Fig. 21. PoD evaluation based on the PDFs of stiffness under 'damaged' scenario (k_12/23_ud indicates the 'undamaged' scenario, k_12/23_d indicates the 'damaged' scenario).

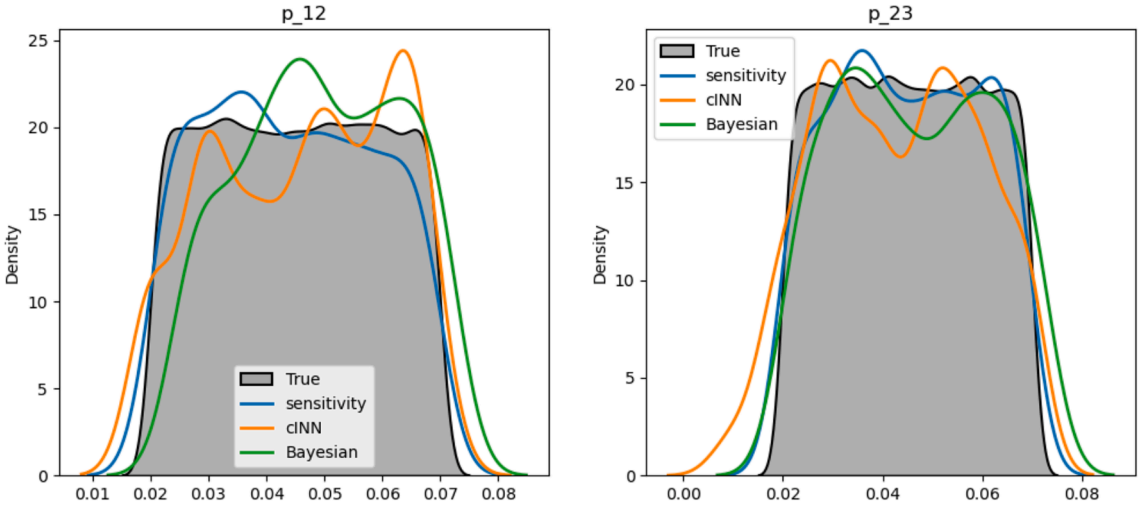


Fig. 22. Comparison for accuracy of model updating results from sensitivity-based, Bayesian, and cINN-based approaches.

Table 9
Means and standard deviations derived by sensitivity-based, Bayesian, and cINN-based approach.

Parameter			Updated result					
	Target value		Sensitivity-based		Bayesian		cINN-based	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
p_{12}	$\mu_1 = 0.045$	$\sigma_1 = 0.0144$	$\mu_1 = 0.0442$	$\sigma_1 = 0.0144$	$\mu_1 = 0.0487$	$\sigma_1 = 0.0141$	$\mu_1 = 0.0473$	$\sigma_1 = 0.0158$
p_{23}	$\mu_2 = 0.045$	$\sigma_2 = 0.0144$	$\mu_2 = 0.0449$	$\sigma_2 = 0.0145$	$\mu_2 = 0.0466$	$\sigma_2 = 0.0153$	$\mu_2 = 0.0453$	$\sigma_2 = 0.0156$

Table 10
Comparison for time efficiency of sensitivity, Bayesian, and cINN-based approaches.

Approaches	Time-consumed[a]
Sensitivity-based	1m17s
Bayesian	26m27s
cINN	Train: 6m41s Inference: 8 s

[a] Computation conducted on a laptop equipped with an Intel(R) Core(TM) Ultra 135U processor.

7. Conclusions

This paper demonstrates the potential of exploiting deep generative models to solve model updating problems by leveraging the conditional Invertible Neural Network (cINN) as a unique data-driven approach to model updating, providing an alternative to the dominant Bayesian sampling-based algorithms and sensitivity-based optimisation methods. Unlike the traditional Bayesian approach, which relies on evaluating complex and intractable likelihood functions, the cINN-based method bypasses this necessity while still producing the posterior distribution of the parameters through its own calibration capabilities.

The cINN's advantages stem from two key features. First, its invertibility enables it to move beyond the role of a surrogate model typically embedded within Bayesian or optimisation frameworks. Instead, the cINN possesses its own calibration mechanism, achieved through a bijective mapping process. Second, the conditional network embedded within the cINN plays a crucial role in handling sparse and noisy observation data. This makes the cINN particularly well-suited for scenarios with limited observational data, ensuring robust model updating even under challenging data conditions.

In addition to its calibration capabilities, the cINN-based approach, as a probabilistic distribution transforming process, has natural advances in uncertainty analysis. The integration of cINN into a multilevel stochastic model updating framework allows for the evaluation of the Probability of Damage (PoD), providing a probabilistic measure of structural health. This enables more informed decision-making under uncertainty, which is critical in applications such as structural health monitoring and damage detection.

While the proposed cINN-based model updating approach demonstrates efficiency and probabilistic inference, several aspects require further investigation. In real-world SHM applications, environmental influences such as seasonal temperature variations and

systematic measurement errors from different test engineers can introduce uncertainties that were not considered in this study. Future work should explore integrating compensation models to address these effects. Additionally, while cINN has the potential to handle high-dimensional problems, its scalability remains to be systematically evaluated, as the case studies in this paper involve only low-dimensional examples. Further research should investigate its performance in large-scale model updating scenarios where hundreds of parameters need calibration, ensuring its efficiency and stability in complex engineering applications.

CRediT authorship contribution statement

Tairan Wang: Writing – original draft, Software, Methodology, Investigation, Data curation. **Sifeng Bi:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Yanlin Zhao:** Validation, Supervision, Methodology, Conceptualization. **Laurent Dinh:** Supervision, Methodology, Investigation. **John Mottershead:** Visualization, Validation, Supervision, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] C.R. Farrar, K. Worden, An introduction to structural health monitoring, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 365 (2006) 303–315, https://doi.org/10.1098/rsta.2006.1928.
[2] V.R. Gharehbaghi, E. Noroozinejad Farsangi, M. Noori, T.Y. Yang, S. Li, A. Nguyen, C. Málaga-Chuquitaype, P. Gardoni, S. Mirjalili, A critical review on structural health monitoring: definitions, methods, and perspectives, Arch Computat Methods Eng 29 (2022) 2209–2235, https://doi.org/10.1007/s11831-021-09665-9.
[3] A. Kamariotis, E. Chatzi, D. Straub, A framework for quantifying the value of vibration-based structural health monitoring, Mech. Syst. Sig. Process. 184 (2023) 109708, https://doi.org/10.1016/j.ymssp.2022.109708.
[4] R. Fuentes, T.P. Howard, M.B. Marshall, E.J. Cross, R.S. Dwyer-Joyce, Observations on acoustic emissions from a line contact compressed into the plastic region, Proceedings of the Institution of Mechanical Engineers, Part J: Journal of Engineering Tribology 230 (2016) 1371–1376, https://doi.org/10.1177/1350650116638590.
[5] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199–222, https://doi.org/10.1016/j.eng.2018.11.030.
[6] Y. Ying, J.H. Garrett, I.J. Oppenheim, L. Soibelman, J.B. Harley, J. Shi, Y. Jin, Toward data-driven structural health monitoring: application of machine learning and signal processing to damage detection, J. Comput. Civ. Eng. 27 (2013) 667–680, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000258.
[7] M.I. Friswell, Damage Identification using Inverse Methods, in: A. Morassi, F. Vestroni (Eds.), Dynamic Methods for Damage Detection in Structures, Springer, Vienna, 2008, pp. 13–66, https://doi.org/10.1007/978-3-211-78777-9_2.
[8] C.-P. Fritzen, D. Jennewein, T. Kiefer, Damage detection based on model updating methods, Mech. Syst. Sig. Process. 12 (1998) 163–186, https://doi.org/10.1006/mssp.1997.0139.
[9] S. Bi, M. Beer, S. Cogan, J. Mottershead, Stochastic model updating with uncertainty quantification: an overview and tutorial, Mech. Syst. Sig. Process. 204 (2023) 110784, https://doi.org/10.1016/j.ymssp.2023.110784.
[10] J.E. Mottershead, M. Link, M.I. Friswell, The sensitivity method in finite element model updating: A tutorial, Mech. Syst. Sig. Process. 25 (2011) 2275–2296, https://doi.org/10.1016/j.ymssp.2010.10.012.
[11] A. Calvi, Uncertainty-based loads analysis for spacecraft: Finite element model validation and dynamic responses, Comput. Struct. 83 (2005) 1103–1112, https://doi.org/10.1016/j.compstruc.2004.11.019.
[12] C. Mares, J.E. Mottershead, M.I. Friswell, Stochastic model updating: Part 1—theory and simulated example, Mech. Syst. Sig. Process. 20 (2006) 1674–1695, https://doi.org/10.1016/j.ymssp.2005.06.006.
[13] J.E. Mottershead, C. Mares, S. James, M.I. Friswell, Stochastic model updating: Part 2—application to a set of physical structures, Mech. Syst. Sig. Process. 20 (2006) 2171–2185, https://doi.org/10.1016/j.ymssp.2005.06.007.
[14] J.L. Beck, L.S. Katafygiotis, Updating models and their uncertainties. I: Bayesian statistical framework, Journal of Engineering Mechanics 124 (1998) 455–461, https://doi.org/10.1061/(ASCE)0733-9399(1998)124:4(455).
[15] J.L. Beck, S.-K. Au, Bayesian updating of structural models and reliability using Markov chain monte carlo simulation, J. Eng. Mech. 128 (2002) 380–391, https://doi.org/10.1061/(ASCE)0733-9399(2002)128:4(380).
[16] K.-V. Yuen, Bayesian Methods for Structural Dynamics and Civil Engineering, John Wiley & Sons, 2010.
[17] E. Patelli, Y. Govers, M. Broggi, H.M. Gomes, M. Link, J.E. Mottershead, Sensitivity or Bayesian model updating: a comparison of techniques using the DLR AIRMOD test data, Arch Appl Mech 87 (2017) 905–925, https://doi.org/10.1007/s00419-017-1233-1.
[18] S. Bi, M. Broggi, M. Beer, The role of the Bhattacharyya distance in stochastic model updating, Mech. Syst. Sig. Process. 117 (2019) 437–452, https://doi.org/10.1016/j.ymssp.2018.08.017.
[19] M. Kitahara, S. Bi, M. Broggi, M. Beer, Distribution-free stochastic model updating of dynamic systems with parameter dependencies, Struct. Saf. 97 (2022) 102227, https://doi.org/10.1016/j.strusafe.2022.102227.
[20] A. Lye, A. Cicirello, E. Patelli, Sampling methods for solving Bayesian model updating problems: A tutorial, Mech. Syst. Sig. Process. 159 (2021) 107760, https://doi.org/10.1016/j.ymssp.2021.107760.
[21] J. Ching, Y.-C. Chen, Transitional Markov chain Monte Carlo method for Bayesian model updating, model class selection, and model averaging, J. Eng. Mech. 133 (2007) 816–832, https://doi.org/10.1061/(ASCE)0733-9399(2007)133:7(816).
[22] A. Rytter, Vibrational Based Inspection of Civil Engineering Structures, (1993).
[23] Y. Huang, C. Shao, B. Wu, J.L. Beck, H. Li, State-of-the-art review on Bayesian inference in structural system identification and damage assessment, Adv. Struct. Eng. 22 (2019) 1329–1351, https://doi.org/10.1177/1369433218811540.
[24] K. Cranmer, J. Brehmer, G. Louppe, The frontier of simulation-based inference, Proceedings of the National Academy of Sciences 117 (2020) 30055–30062, https://doi.org/10.1073/pnas.1912789117.

[25] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, arXiv.Org (2013). https://arxiv.org/abs/1312.6114v11 (accessed October 7, 2024).
[26] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks (2014), https://doi.org/10.48550/arXiv.1406.2661.
[27] L. Dinh, D. Krueger, Y. Bengio, NICE: Non-linear Independent Components Estimation, (2015). http://arxiv.org/abs/1410.8516 (accessed November 6, 2023).
[28] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using Real NVP, (2017). http://arxiv.org/abs/1605.08803 (accessed November 5, 2023).
[29] D.P. Kingma, P. Dhariwal, Glow: Generative Flow with Invertible 1x1 Convolutions, (2018). https://doi.org/10.48550/arXiv.1807.03039.
[30] J.M. Tomczak, Deep Generative Modeling, Springer International Publishing, Cham, 2022 https://doi.org/10.1007/978-3-030-93158-2.
[31] L. Ardizzone, C. Lüth, J. Kruse, C. Rother, U. Köthe, Guided Image Generation with Conditional Invertible Neural Networks, (2019). http://arxiv.org/abs/1907.02392 (accessed November 10, 2023).
[32] S.T. Radev, U.K. Mertens, A. Voss, L. Ardizzone, U. Köthe, BayesFlow: learning complex stochastic models with invertible neural networks, IEEE Trans. Neural Networks Learn. Syst. 33 (2022) 1452–1466, https://doi.org/10.1109/TNNLS.2020.3042395.
[33] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E.W. Pellegrini, R.S. Klessen, L. Maier-Hein, C. Rother, U. Köthe, Analyzing Inverse Problems with Invertible Neural Networks, (2019). https://doi.org/10.48550/arXiv.1808.04730.
[34] G.A. Padmanabha, N. Zabaras, Solving inverse problems using conditional invertible neural networks, J. Comput. Phys. 433 (2021) 110194, https://doi.org/10.1016/j.jcp.2021.110194.
[35] J. Zeng, M.D. Todd, Z. Hu, Probabilistic damage detection using a new likelihood-free Bayesian inference method, J Civil Struct Health Monit 13 (2023) 319–341, https://doi.org/10.1007/s13349-022-00638-5.
[36] P. Noever-Castelos, L. Ardizzone, C. Balzani, Model updating of wind turbine blade cross sections with invertible neural networks, Wind Energy 25 (2022) 573–599, https://doi.org/10.1002/we.2687.
[37] S. Bi, M. Broggi, P. Wei, M. Beer, The Bhattacharyya distance: Enriching the P-box in stochastic sensitivity analysis, Mech. Syst. Sig. Process. 129 (2019) 265–281, https://doi.org/10.1016/j.ymssp.2019.04.035.
[38] I. Isnardi, E. Menga, J.E. Mottershead, S. Fichera, An experimental study on the stochastic model updating of a structure with irreducible parameter variability and fixed but unknown hyperparameters, Mech. Syst. Sig. Process. 200 (2023) 110597, https://doi.org/10.1016/j.ymssp.2023.110597.