Journal of Civil Structural Health Monitoring (2023) 13:319–341
https://doi.org/10.1007/s13349-022-00638-5

ORIGINAL PAPER

Probabilistic damage detection using a new likelihood-free Bayesian inference method

Jice Zeng[1] · Michael D. Todd[2] · Zhen Hu[1]

Received: 30 July 2022 / Accepted: 8 October 2022 / Published online: 19 October 2022

Abstract

Bayesian inference plays a vital role in Structural Health Monitoring (SHM) by assessing structural integrity through probabilistic model updating using monitoring data. A crucial component in Bayesian inference is the evaluation of the likelihood function. For some situations, the likelihood function is not available in closed form or is analytically intractable, due to either a computationally intractable forward model solution or lack of measurements for some inputs. While likelihood-free Bayesian inference methods such as approximate Bayesian computation have been proposed to tackle this issue, the required computational effort is high, and the accuracy relies on similarity criteria, which makes these methods unsuitable for online monitoring. This study investigates a novel likelihood-free and computationally efficient Bayesian inference method, named BayesFlow, for probabilistic damage inference in SHM through model updating. It consists of a training phase and an online monitoring phase. In the training phase, BayesFlow approximates the posterior distributions of structural parameters by jointly training a conditional invertible neutral network (cINN) and a summary network. The cINN connects structural parameters with a latent of the posterior distribution. The summary network automatically learns the maximally informative statistics for model updating from data of output variables rather than hand-crafted features. In the online monitoring phase, BayesFlow directly predicts the posterior distribution for any given observations, without computing any likelihood or evaluating the forward prediction model, and thereby allows for real-time monitoring. Two benchmark examples, including an 18-story steel frame and a concrete building frame, are used to demonstrate the proposed method. Results comparison of the proposed method and the Differential Evolutionary Adaptive Metropolis (DREAM) sampling method demonstrates advantages of the proposed method in terms of both accuracy and computational efficiency.

Keywords Structural health monitoring (SHM) · Model updating · BayesFlow · Invertible neural network · Probabilistic damage detection

1 Introduction

Structural health monitoring (SHM) plays an important role in ensuring safety and serviceability of civil infrastructure. The general paradigm involves periodic or continuous inspection and/or data collection with in situ monitoring systems from which information regarding structural health is mined that can inform structural integrity or even maintenance actions [1, 2]. One widely used class of SHM approaches is finite element model updating (FEMU), which adjusts finite element model parameters by minimizing the discrepancy between model predictions and the measured counterparts [3, 4]. The updated model gives engineering practitioners a variety of benefits in model-based tasks, such as damage detection, risk and reliability assessment, structural control, and failure prognostics [5, 6]. An entire other class of SHM approaches involves data-driven paradigms, using both unsupervised or supervised learning strategies, but such strategies are challenging in most civil applications due to the sparseness or incompleteness of data as well as observations of failure/limit states.

FEMU methods in general can be categorized as deterministic or probabilistic. Deterministic methods formulate

✉ Zhen Hu
zhennhu@umich.edu

[1] Department of Industrial and Manufacturing Systems Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA

[2] Department of Structural Engineering, University of California San Diego, La Jolla, San Diego, CA 92093-0085, USA

FEMU as an optimization problem that targets the goodness of fit between measurements and model-derived responses using optimization algorithms such as metaheuristic algorithms [7, 8]. The main drawback of deterministic methods is that they only give point estimates without any sense of confidence or uncertainty bounds. For models with nonlinear responses, the ill-posedness of inverse problems and the presence of various uncertainty sources in the simulation environment introduce significant challenges to classical optimization-based FEMU [9, 10]. On the other hand, probabilistic methods overcome these limitations by updating uncertain model parameters as a distribution function by considering the sources of uncertainty in the process, such as model form uncertainty and measurement error [11]. Among various probabilistic methods, one of the most popular methods used for SHM applications is Bayesian model updating, which has been extensively studied in the context of probabilistic damage detection and system identification [12–14]. In Bayesian model updating, the prior or existing knowledge (subjective information, or plausibility) and experimental/observation/monitoring data (new information) are combined to estimate the posterior distribution of uncertain model parameters using Bayes' theorem [15]. Due to the capability of accounting various uncertainty sources in model updating, Bayesian methods have rapidly become a promising tool for parameter identification and damage assessment of complex engineering structures in SHM community.

The most essential component of Bayesian inference is the likelihood function, which represents the probability of observing measurements conditioned upon a forward predictive model in the presence of uncertainty. According to the way that the likelihood function is used, Bayesian model updating can be roughly grouped the traditional likelihood-based approaches and the likelihood-free approaches. The likelihood-based approach, such as Markov chain Monte Carlo (MCMC) simulation, requires the evaluation of the likelihood function given in analytical or numerical form. For some situations, however, the likelihood function is: (1) computationally prohibitive to evaluate, due to either the involvement of a computationally expensive forward model or the need to solve high-dimensional integrals with the consideration of various uncertainty sources [16, 17]; or (2) analytically or numerically intractable due to model complexity.

To address computational challenges with the forward model, various surrogate modeling methods have been developed using either reduced-order models [18] or metamodels, such as Gaussian process regression [19], polynomial chaos expansion [20], artificial neural network [21], etc. For SHM applications where structures respond with ambient vibration, the application of surrogate model in model updating, however, is limited to the use of scalar-valued or time-averaged data, such as modal data including natural frequency and modal assurance criterion (mode shape) extracted from time series data [22]. Multiple surrogate models are needed to relate a certain type of modal data with selected model parameters, which may increase the computational cost [23]. The performance of model updating is tied to how well modal data are identified and what modal data are adopted. The direct usage of output-only time series data for SHM under ambient vibration has been largely ignored in surrogate-based model updating. Furthermore, surrogate models may not lead to remarkable decreases in computational time, since the fundamental limitation of evaluating the likelihood function numerous times remains.

Motivated by tackling the challenge that the likelihood function is often intractable for Bayesian inference, likelihood-free approaches have been developed and received considerable attention. Likelihood-free approaches directly sample from, rather than directly evaluate, a likelihood function to approximate the posterior distribution. The most well-developed approach in the context of likelihood-free cases is approximate Bayesian computation (ABC) [24]. In ABC, model parameters are repeatedly drawn from a prior distribution, and synthetic datasets are generated by running the forward model with those parameter samples. If the similarity between the simulated data and the actual observation satisfies a certain user-specified threshold, the corresponding parameters 'survive' as the samples of the target posterior or are otherwise discarded. ABC is particularly useful to treat problems with intractable likelihood functions and has grown in popularity in SHM. Several sampling methods have been combined with ABC to improve the accuracy of parameter estimation for complex systems. Fang et al. [25] incorporated ABC with Metropolis Hastings sampling and response surface method to achieve fast and probabilistic damage detection for a reinforced concrete beam. Fernández et al. [26] developed a novel gradient-free method based on ABC and subset simulation, which was experimentally verified by a composite material with fatigue damage. Ritto et al. [27] integrated reinforcement learning with ABC to realize efficient model selection and parameter updating for a nonlinear dynamic system. Kitahara et al. [28] also developed an ABC model updating framework incorporating staircase random variables and Bhattacharyya distance for stochastic model updating and uncertainty quantification. In addition, Barros et al. [29] proposed an adaptive ABC method to sequentially identify hyper-parameters for non-linear structural model updating. Fang and Chen [30] introduced a gray Bayesian model updating strategy based on ABC and population Monte Carlo sampler, which was applied for multi-damage detection on laboratory-scale beam.

However, ABC has disadvantages in some respects. First, the tolerance level of the "accept-reject" mechanism in ABC greatly affects the approximation accuracy of the posterior distribution. Strict tolerance levels result in a desirable accuracy level while the required computational cost is amplified substantially due to a high rejection rate, since many candidate samples get rejected, and many simulations are required to ensure enough samples will span the posterior distribution. On the other hand, a large tolerance level could increase sample efficiency at the expense of inference quality. Second, the entire estimation procedure in ABC needs to be repeated from scratch for any new measurement data, which restricts ABC's application to a single dataset or up to a few data points in the independent and identically distributed case [31]. Third, ABC also suffers from data dimensionality challenges, as the required number of simulations increases dramatically with dimensionality. These disadvantages can make ABC unsuitable for online SHM applications demanding accurate inference in reasonable time frames [32].

The goal of this paper is to address limitations of current likelihood-free approaches for probabilistic damage detection, by applying a novel likelihood-free and computationally efficient Bayesian inference method, named BayesFlow developed by Radev et al. [31], to Bayesian model updating in SHM. In contrast to other likelihood-free methods (e.g., ABC), BayesFlow successfully realizes amortized inference, in which the entire parameter estimation is split into an upfront training phase that is computationally intensive and a subsequent inference phase that is very quick to execute. It is a fully likelihood-free approach that directly estimates the posterior distribution without repeatedly evaluating the likelihood function in the inference phase. BayesFlow encompasses two separate neural networks—a summary network and an inference network—to complete the task of parameter inference. The summary network is responsible for reducing data dimensionality from potentially large time series datasets to a fixed-size vector. Unlike traditional approaches that use summary statistics manually pre-selected by the user, BayesFlow automatically learns the maximally informative statistics from the raw data. The inference network is executed as a conditional invertible neural network (cINN), which predicts the posterior distribution efficiently for any given measurements after training. These two networks are jointly trained and aligned well for parameter inference given synthetic data generated from a forward model. The technical details will be explained in detail in Sect. 3. Another appealing feature in BayesFlow is that it allows for Bayesian inference using different sizes of dataset using a single trained model. This property is valuable in practice, since the number of measurements in damage detection may vary with time duration and measurement circumstances. The above features make BayesFlow a promising solution to the drawbacks of current likelihood-free approaches for Bayesian model updating-based damage detection. This work attempts to reveal this promising potential of BayesFlow, and specifically adapt BayesFlow for the purpose of probabilistic damage detection in civil infrastructures. To the best of our knowledge, BayesFlow has not yet been applied in structural model updating and probabilistic damage detection in SHM applications.

The remainder of this paper is organized as follows. Background of damage detection using Bayesian inference and likelihood-free methods are introduced in Sect. 2. Section 3 presents the fundamentals of BayesFlow method and discusses its application in damage detection. Two benchmark examples including an 18-story steel shear frame and a concrete frame building are utilized to demonstrate the capability of BayesFlow in damage detection in Sect. 4. A comparative study between BayesFlow and the existing method is also investigated in this section. Finally, conclusions are drawn in Sect. 5.

2 Background

This section first provides a brief introduction of damage detection using Bayesian model updating. Following that, the current methods and their limitations are discussed.

2.1 Bayesian model updating for structural damage detection

In Bayesian model updating used for structural damage detection, measurement data from a physical system are used to update its numerical representation (e.g., FE model) to estimate the structural damage characterized by uncertain model parameters $\boldsymbol{\theta}$. Let $\mathbf{y}_k = \eta(\boldsymbol{\theta}, \mathbf{u}_{1:k})$ be an FE model, where $\mathbf{y}_k \in \mathbb{R}^{N_Y \times 1}$ are the model outputs at time step k (i.e., time t_k), N_Y is the number of outputs, $\mathbf{u}_{1:k} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \cdots, \mathbf{u}_k^T] \in \mathbb{R}^{(N_u \times k) \times 1}$ are the input excitations over the past k time steps, and N_u is the number of input excitation variables. The FE model can be related to measurements or observations $\mathbf{y}_{o,k} \in \mathbb{R}^{N_Y \times 1}$ as follows,

$$\begin{aligned} \mathbf{y}_{o,k} &= \mathbf{y}_k + \delta(\mathbf{u}_{1:k}) + \boldsymbol{\varepsilon}_k, \\ &= \eta(\boldsymbol{\theta}, \mathbf{u}_{1:k}) + \delta(\mathbf{u}_{1:k}) + \boldsymbol{\varepsilon}_k, \end{aligned} \tag{1}$$

where $\delta(\mathbf{u}_{1:k})$ is the model discrepancy of the FE model, $\boldsymbol{\varepsilon}_k \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ is the Gaussian noise term with zeros means and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{N_Y \times N_Y}$ at t_k, $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma}_k = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{N_Y}^2 \end{pmatrix} \in \mathbb{R}^{N_Y \times N_Y}, \tag{2}$$

in which $\sigma_i^2, \ i = 1, \cdots, N_Y$ are the standard deviations of observation/measurement noise of the i-th response.

If both the input excitation and the outputs are measured, the uncertain model parameters $\boldsymbol{\theta}$ can be estimated or updated using Bayes' theorem as follows,

$$f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}_{o,\,1:k},\ \mathbf{u}_{1:k}) = \frac{f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\,1:k}|\boldsymbol{\theta},\ \mathbf{u}_{1:k})f_{\boldsymbol{\theta}}(\boldsymbol{\theta})}{\int f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\,1:k}|\boldsymbol{\theta},\ \mathbf{u}_{1:k})f_{\boldsymbol{\theta}}(\boldsymbol{\theta})\mathbf{d}\boldsymbol{\theta}} \propto f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\,1:k}|\boldsymbol{\theta},\ \mathbf{u}_{1:k})f_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \tag{3}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$ which reflects existing knowledge, engineers' opinions and expected physical meaning (and can be expressed as an uniformed prior if desired), $\mathbf{y}_{o,\,1:k} = [\mathbf{y}_{o,\,1},\ \cdots,\ \mathbf{y}_{o,\,k}] \in \mathbb{R}^{N_Y \times k}$ are the observations/measurements of the outputs from t_1 to t_k, and $f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\,1:k}|\boldsymbol{\theta},\ \mathbf{u}_{1:k})$ is the likelihood function of observing $\mathbf{y}_{o,\,1:k}$ for given $\boldsymbol{\theta}$ and measurements of $\mathbf{u}_{1:k}$.The likelihood function reflects the degree of belief ("plausibility") that the model, characterized by the parameter vector $\boldsymbol{\theta}$. explains the actual observations.

The posterior probability density function $f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}_{o,\,1:k},\ \mathbf{u}_{1:k})$ can be estimated using MCMC sampling methods such as Metropolis Hasting (MH) [33], delayed rejection and adaptive Metropolis (DRAM) [34], Differential Evolutionary Adaptive Metropolis (DREAM) [35], sequential Monte Carlo simulation (SMC) [36], etc.

2.2 Current methods and limitations

In structural dynamics, external excitations $\mathbf{u}_{1:k}$ can be either measured or unmeasured depending on circumstances. When the $\mathbf{u}_{1:k}$ are measured, structural damage detection can be performed directly using Eq. (3) and the methods mentioned above. For some situations, however, the $\mathbf{u}_{1:k}$ are unmeasured. For instance, for damage detection under ambient vibration, the external ambient excitation is unknown and unmeasured, and is usually assumed to be broadband Gaussian white noise [37]. Damage detection under ambient vibration has received considerable interests since it is economically viable and commercially sustainable, particularly for large civil infrastructure systems [38]. The advances and development in sensor technology along with powerful data acquisition systems make it possible to collect vibration data under ambient vibration for SHM applications. The major benefit of using ambient vibration data, e.g., wind, traffic, or human induced vibrations, against that of forced vibrations is that any special, expensive, and/or intrusive excitation equipment are not required, and the system doesn't have to be taken out of service for the specialized controlled-excitation tests [39].

When the input excitations are not measured in this scenario, Eq. (3) needs to be modified as follows,

$$\begin{aligned} f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}_{o,\,1:k}) &= \int f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}_{o,\,1:k},\ \mathbf{u}_{1:k})f_{\mathbf{u}}(\mathbf{u}_{1:k})\mathbf{du}_{1:k}, \\ &\propto \int f_{\mathbf{y}|\boldsymbol{\theta}}(\mathbf{y}_{o,\,1:k}|\boldsymbol{\theta},\ \mathbf{u}_{1:k})f_{\mathbf{u}}(\mathbf{u}_{1:k})\mathbf{du}_{1:k}f_{\boldsymbol{\theta}}(\boldsymbol{\theta}), \end{aligned} \tag{4}$$

where $f_{\mathbf{u}}(\mathbf{u}_{1:k})$ is the join probability density function (PDF) of $\mathbf{u}_{1:k}$. The goal of Eq. (4) is to account for the uncertainty in the unmeasured excitation during the evaluation of likelihood function and Bayesian model updating.

The consideration of unmeasured excitations or other unmeasured uncertainty sources significantly increases the difficulty of evaluating the likelihood function and makes the likelihood function analytically intractable and computationally expensive to compute. As a result, MCMC sampling methods, as representatives of current popular Bayesian inference methods, cannot be directly employed for damage detection in the time domain using only vibration responses measured under ambient vibration. As mentioned in Sect. 1, ABC method or its variants provide a potential solution to the aforementioned challenges with the likelihood function using likelihood-free inference methods. These likelihood-free methods, however, require a user-defined tolerance level that greatly affects the accuracy of the approximated posterior. A more accurate approximation usually requires a high rejection rate. Furthermore, the entire estimation procedure needs to be repeated from scratch for any given new dataset. They also suffer from curse of dimensionality [32].

In the context of damage detection with unmeasured excitation, another commonly used approach is to convert time series measurement data from ambient excitation into frequency domain data (e.g., modal data including natural frequency and mode shapes) and then apply MCMC methods. Modal parameters can be identify by stochastic system identification methods for output-only measurement conditions, using Eigensystem Realization Algorithm [40], stochastic subspace identification [41], or Bayesian operational modal identification [42]. Based on this, the conventional MCMC-based Bayesian model updating methods may be applied in the frequency domain to perform damage detection using modal data. Typically, MCMC-based Bayesian methods using modal data give a posterior PDF of model parameters $\boldsymbol{\theta}$ given the measured data $\mathbf{y}_{o,\,1:k}$ as follows [43, 44],

$$f_{\boldsymbol{\theta}|\mathbf{y}}(\boldsymbol{\theta}|\mathbf{y}_{o,1:k}) = c_0 \exp\left(-\frac{1}{2}J(\boldsymbol{\theta})\right), \tag{5}$$

where c_0 is a constant normalizing the posterior PDF, the measure of fit function $J(\boldsymbol{\theta})$ is given by

$$J(\boldsymbol{\theta}) = \sum_{m=1}^{N_m} \left(w_{F_m}(\tilde{F}_m - F_m(\boldsymbol{\theta}))^2 + w_{\phi_m} \left\| \tilde{\boldsymbol{\psi}}_m - \boldsymbol{\psi}_m(\boldsymbol{\theta}) \right\|^2 \right), \tag{6}$$

where N_m is the total number of modes to be considered in model updating; $\|\cdot\|$ is Euclidean norm; w_{F_m} and w_{ϕ_m} are chosen weightings for the m-th measured frequency and mode shapes; $\tilde{F}_m$ and $\tilde{\boldsymbol{\psi}}_m$ are respectively the m-th measured frequency and mode shape that are identified from $\mathbf{y}_{o,1:k}$; $F_m(\boldsymbol{\theta})$ and $\boldsymbol{\psi}_m(\boldsymbol{\theta})$ are respectively the m-th FE model-derived frequency and mode shape given $\boldsymbol{\theta}$ that can be calculated using commercial software, e.g., ANSYS, or characteristic equations $(\mathbf{K} - \boldsymbol{\lambda}\mathbf{M})\boldsymbol{\psi} = \mathbf{0}$, where $\mathbf{K}$ and $\mathbf{M}$ are global stiffness and mass matrix, respectively; and $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$ are eigenvalues and eigenvectors, respectively.

It is worth noting that the features contained in modal data in Eq. (6) have been widely used as damage indicators for structural health assessment. For instance, Mustafa and Matsumoto [45] proposed a novel Bayesian model updating framework and performed damage detection on an existing truss bridge using modal data. The partial fracture on diagonal member was identified. Ding et al. [46] proposed a new damage identification method based on Jaya algorithm and Bayesian inference with modal data, which was validated by a pre-stressed concrete bridge. Yang and Lam [47] also developed adaptive sequential Monte Carlo for damage detection using Bayesian model updating and modal data. The methodology was verified by a laboratory shear building and transmission tower. Zhou et al. [48] investigated Bayesian model updating for incremental damage detection on an actual steel truss bridge. The natural frequency and mode shape were considered in the updating process. Zeng and Kim [49] presented a new Bayesian model updating with mass addition, two sets of modal data were used to perform probabilistic damage detection for a laboratory shear building. A comprehensive review on the application of modal data for model updating and damage detection can be found in [50].

Although numerous research efforts using MCMC sampling methods and modal data for SHM applications have been reported, there are still some limitations. Due to unmeasured excitations, the raw vibration responses need to be pre-processed with extra effort for modal data. Despite mature and sophisticated modal identification methods, it still inevitably leads to identification error in modal data due to low-level data quality, deficiency of identification methods, and weak ambient excitation. In addition, modal data usually contains limited information, i.e., only the first few modes are accurately identified. The error in modal data will also be propagated to errors in parameter inference for damage detection. In addition, in most cases, the posterior PDF is formulated based on the prediction error (i.e., frequency error and mode shape error, as shown in Eq. (6)), which is usually assumed to be independent identically distributed Gaussian errors with zero-mean and constant variance. This assumption, however, may be questionable and lead to a biased parameter identification [51]. Finally, in MCMC sampling methods using modal data, only uncertainty from measurement noise (e.g., w_{F_m} and w_{ϕ_m} in Eq. (6)) is accounted for in parameter inference. There is a consensus that modeling errors/bias are often the most significant source of uncertainty in modeling but usually ignored, which underestimates the uncertainty and may not guarantee the reliability of parameter estimation [52].

Furthermore, for both ABC and frequency-domain methods, they must be implemented from scratch whenever a new set of measurements is available. This makes damage detection using Bayesian model updating computationally expensive and not suitable for online model updating. The above discussed damage detection under ambient vibration is just one example. In reality, even for systems with measured excitations, the likelihood functions could be numerically intractable and computationally expensive due to either high model complexity (e.g., multiple models connected in a hierarchical manner) or the influence of many sources of uncertainty, such as non-Gaussian and dependent measurement noise, model form uncertainty, etc. [52] A new likelihood-free inference method is needed to overcome the limitations of the current methods for probabilistic damage detection using Bayesian model updating.

Motivated by enhancing the accuracy and efficiency of damage detection using Bayesian model updating, a novel likelihood-free and computationally efficient Bayesian inference, named BayesFlow, is introduced in the next section.

3 Damage detection using a new likelihood-free Bayesian inference method

This section first provides an introduction of normalizing flows and conditional invertible neural network (cINN). Following that, theories of BayesFlow are presented. Finally, this section discusses the application of BayesFlow for damage detection using Bayesian model updating.

3.1 Normalizing flows

Let $\boldsymbol{\theta} \in \mathbb{R}^N$ be random variables with complex and irregular PDF $f_\theta(\cdot) : \mathbb{R}^N \to \mathbb{R}$, and $\mathbf{Z} \in \mathbb{R}^N$ be a multivariate Gaussian distribution with PDF $f_{\mathbf{Z}}(\mathbf{z}) \in \mathbb{R}$. There are two types of mapping between the two distributions, namely generative direction and normalizing direction. In the generative direction, we first sample $\mathbf{z}$ from $f_{\mathbf{Z}}(\mathbf{z})$ and then use generator $\boldsymbol{\theta} = \mathbf{g}(\mathbf{z})$, where $\mathbf{g}(\cdot)$ is an invertible function, to obtain samples of $\boldsymbol{\theta}$. Let $\mathbf{h}(\cdot) = \mathbf{g}^{-1}(\cdot)$ be the inverse of $\mathbf{g}(\cdot)$ such that $\mathbf{z} = \mathbf{g}^{-1}(\boldsymbol{\theta}) = \mathbf{h}(\boldsymbol{\theta})$, in the normalizing direction, $\mathbf{h}(\cdot)$ maps the complex and irregular distribution of $\boldsymbol{\theta}$ to a multivariate Gaussian distribution [53].

Based on the above definitions, the two PDFs are related to each other as [53]

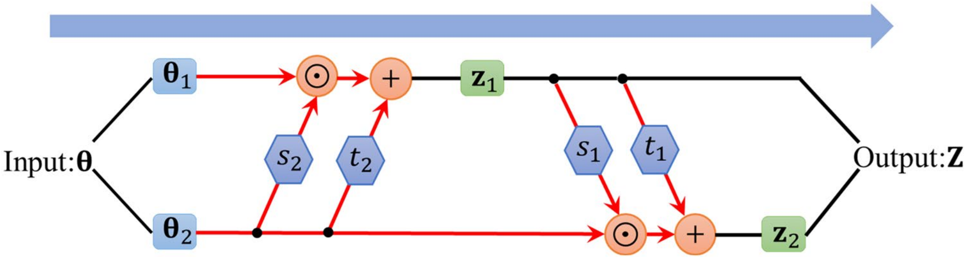


Fig. 1 The forward transformation in ACL (normalizing direction)

$$f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = f_{\mathbf{z}}(\mathbf{z})\left|\det\left(\frac{\partial \mathbf{h}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\right| = f_{\mathbf{z}}(\mathbf{z})\left|\det\left(\frac{\partial \mathbf{g}(\mathbf{h}(\boldsymbol{\theta}))}{\partial \mathbf{z}}\right)\right|^{-1}. \tag{7}$$

In practice, it is difficult to construct complicated invertible functions for the above-described non-linear bijective transformation. To overcome this challenge, Dinh et al. [54] express $\mathbf{h}(\cdot)$ as a set of M bijective functions as $\mathbf{h}(\cdot) = \mathbf{h}_1(\cdot)\circ \cdots \circ\mathbf{h}_{M-1}(\cdot)\circ\mathbf{h}_M(\cdot)$ and that $\mathbf{g}_j(\cdot) = \mathbf{h}_j^{-1}(\cdot),\ j = 1, \cdots, M$ exist. The resulting $\mathbf{g}(\cdot) = \mathbf{g}_1(\cdot)\circ \cdots \circ\mathbf{g}_{M-1}(\cdot)\circ\mathbf{g}_M(\cdot)$ is also bijective. Based on this expression, they proposed the concept of *affine coupling layers* (ACL) as $\mathbf{h}_j(\cdot),\ j = 1,\ \cdots,\ M$ to achieve the invertible mapping between inputs and outputs.

Each ACL implements an inverse non-linear transformation, such as a general forward mapping $f_{ACL}(\cdot)$ and an inverse mapping $f_{ACL}^{-1}(\cdot)$. In each ACL, four internal functions or subnetworks are embedded, denoted as $s_1(\cdot),\ s_2(\cdot),\ t_1(\cdot),\ t_2(\cdot)$, as shown in Figs. 1 and 2. The four subnetworks do not need to be inverted and can be selected as any arbitrary neural networks, such as fully connected neural networks. A single ACL splits the input and output vectors $\boldsymbol{\theta}$ and $\mathbf{z}$ into two halves $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\ \boldsymbol{\theta}_2)$ and $\mathbf{z} = (\mathbf{z}_1,\ \mathbf{z}_2)$, respectively. The forward transformation is shown in Fig. 1 and realized by the following operations [54]

$$\mathbf{z}_1 = \boldsymbol{\theta}_1 \odot \exp(s_2(\boldsymbol{\theta}_2)) + t_2(\boldsymbol{\theta}_2), \tag{8}$$

$$\mathbf{z}_2 = \boldsymbol{\theta}_2 \odot \exp(s_1(\mathbf{z}_1)) + t_1(\mathbf{z}_1), \tag{9}$$

where $\odot$ is the element-wise multiplication.

Similarly, the outputs $\mathbf{z} = (\mathbf{z}_1,\ \mathbf{z}_2)$ are concatenated and inversely pass through the ACL. As illustrated in Fig. 2, the inverse operations are given by [54]

$$\boldsymbol{\theta}_1 = (\mathbf{z}_1 - t_2(\boldsymbol{\theta}_2)) \odot \exp(-s_2(\boldsymbol{\theta}_2)), \tag{10}$$

$$\boldsymbol{\theta}_2 = (v_2 - t_1(v_1)) \odot \exp(-s_1(v_1)). \tag{11}$$

The simple mathematical expression of Jacobian in ACL (upper or lower triangle matrix) makes the determinant of Jacobian given in Eq. (7) computationally cheap to evaluate, and thus facilitates bijective transformation of the distributions.

3.2 Conditional invertible neural network (cINN) architecture

By taking observations $\mathbf{y}_o$ as an additional input of neural networks $s_1(\cdot),\ s_2(\cdot),\ t_1(\cdot)$, and $t_2(\cdot)$ in the original ACL, as shown in Fig. 3, a conditional ACL (cACL) can be constructed. For the forward transformation, the operations given in Eqs. (8) and (9) become

$$\mathbf{z}_1 = \boldsymbol{\theta}_1 \odot \exp(s_2(\boldsymbol{\theta}_2,\ \mathbf{y}_0)) + t_2(\boldsymbol{\theta}_2,\ \mathbf{y}_0), \tag{12}$$

$$\mathbf{z}_2 = \boldsymbol{\theta}_2 \odot \exp(s_1(\mathbf{z}_1,\ \mathbf{y}_0)) + t_1(\mathbf{z}_1,\ \mathbf{y}_0). \tag{13}$$

The inverse transformation given in Fig. 2 and Eqs. (10) and (11) can be revised accordingly for the cACL. By sequentially stacking multiple cACLs together to establish a sufficient neural network, it allows for a non-linear bijective mapping between a complex distribution $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_o)$ and a

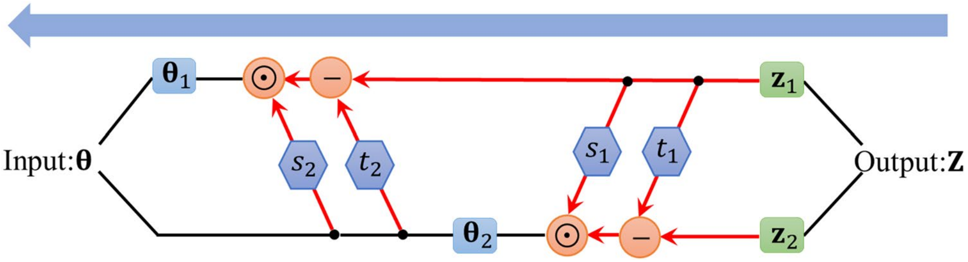


Fig. 2 The inverse transformation in ACL (generative direction)

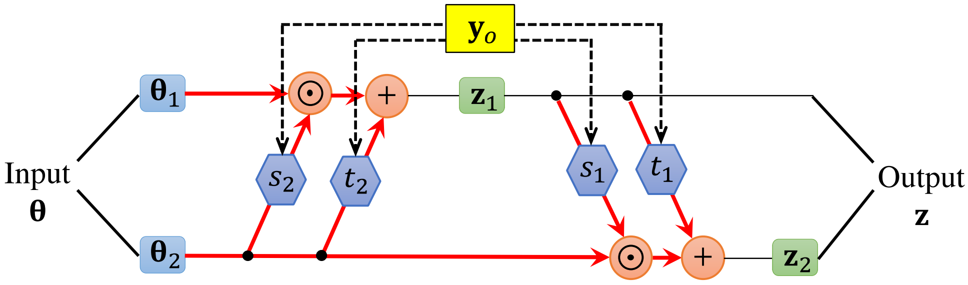


Fig. 3 The structure of cACL for forward transformation

multivariate Gaussian distribution $f_{\mathbf{Z}}(\mathbf{z})$. The resulting network is called a conditional invertible neural network (cINN) [55]. In the entire cINN, the output of each cACL serves as the input of the next one. cINN can be considered as an inverse surrogate model. In summary, the mappings between $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_o)$ and a multivariate Gaussian PDF $f_{\mathbf{Z}}(\mathbf{z})$ in cINN can be realized using an invertible function $\mathbf{z} = \mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \mathbf{y}_o)$ with model parameters $\boldsymbol{\omega}$ for normalizing direction transformation, and its inverse function $\boldsymbol{\theta} = \mathbf{h}_{\boldsymbol{\omega}}^{-1}(\mathbf{z};\ \mathbf{y}_o)$ for generative direction transformation. The latent variable $\mathbf{z}$ following a multivariate Gaussian distribution plays a crucial role in cINN.

3.3 BayesFlow

BayesFlow is built upon normalizing flow-based theory [55, 56] and cINN described above. It is proposed by Radev and co-workers for neurocognitive and epidemiology models [31]. The goal of BayesFlow is to approximate posterior distribution $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})$ of $\boldsymbol{\theta}$ for any given observations $\mathbf{y}_{1:T}$ using cINN, where $\mathbf{y}_{1:T} = (\mathbf{y}_1,\ \mathbf{y}_2,\ \cdots,\ \mathbf{y}_T)$ and $\mathbf{y}_i,\ \forall i = 1,\ \cdots,\ T$ is the i-th vector of observations. In addition to cINN, BayesFlow introduces and jointly trains a summary network along with cINN to deal with high-dimensional time series data in inference.

The summary network is essentially a preprocessing step for simulated or measured data prior to training cINN. Measured raw data (i.e., $\mathbf{y}_{1:T}$) is summarized or filtered using summary network to a fixed-size and low-dimensional vector. Mathematically, the summary network can be represented as

$$\tilde{\mathbf{y}} = \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}), \tag{14}$$

where $\varphi_{\boldsymbol{\gamma}}(\cdot)$ is the summary network with parameters $\boldsymbol{\gamma}$ and $\tilde{\mathbf{y}}$ is the summarized feature from the network which will be used as $\mathbf{y}_0$ in the inference network (i.e., cINN). The choice of the summary network depends on the properties of measured data $\mathbf{y}_{1:T}$. For example, a bidirectional long short-term memory (LSTM) [57] as a summary network is well tailored for time series data, since LSTM network is designed to deal with sequential measurements with long-term memory. Another preferred summary network is a 1D fully connected convolutional neural network (CNN), which has been widely adopted to learn summary statistics of temporal responses [58].

To jointly train the inference network $\mathbf{h}_{\boldsymbol{\omega}}(\cdot)$ (i.e., cINN) and the summary network $\varphi_{\boldsymbol{\gamma}}(\cdot)$ for the mapping between $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})$ and $f_{\mathbf{Z}}(\mathbf{z})$, BayesFlow estimates neural network model parameters $\boldsymbol{\omega}$ and $\boldsymbol{\gamma}$ by minimizing the expected Kullback–Leibler (KL) divergence between the target and the approximated posteriors for observations $\mathbf{y}_{1:T}$ as below [31]

$$\begin{aligned}
\hat{\boldsymbol{\gamma}},\ \hat{\boldsymbol{\omega}} &= \underset{\boldsymbol{\gamma},\ \boldsymbol{\omega}}{\arg\min}\, \mathrm{E}_{f_{\mathbf{Y}_{1:T}}(\mathbf{y}_{1:T})}\left[\mathrm{KL}\left[f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})\middle\|\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\right]\right],\\
&= \underset{\boldsymbol{\gamma},\ \boldsymbol{\omega}}{\arg\min}\, \mathrm{E}_{f_{\mathbf{Y}}(\mathbf{y}_{1:T})}\left[\mathrm{E}_{f_{\boldsymbol{\theta}|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})}\left[\log\left\{f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})\right\} - \log\{\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\}\right]\right],\\
&= \underset{\boldsymbol{\gamma},\ \boldsymbol{\omega}}{\arg\max}\, \mathrm{E}_{f_{\mathbf{Y}}(\mathbf{y}_{1:T})}\left[\mathrm{E}_{f_{\boldsymbol{\theta}|\mathbf{Y}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})}\left[\log\{\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\}\right]\right],\\
&= \underset{\boldsymbol{\gamma},\ \boldsymbol{\omega}}{\arg\max} \iint f_{\boldsymbol{\theta},\mathbf{Y}}(\mathbf{y}_{1:T},\ \boldsymbol{\theta})\log\{\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\}\mathbf{d}\boldsymbol{\theta}\mathbf{d}\mathbf{y}_{1:T},
\end{aligned} \tag{15}$$

where $f_{\mathbf{y}_{1:T}}(\mathbf{y}_{1:T})$ is the PDF of $\mathbf{y}_{1:T}$, E[·] is expectation operator, $\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))$ is the estimated posterior of $\boldsymbol{\theta}$ for given parameters $\boldsymbol{\omega}$ and $\boldsymbol{\gamma}$ of the cINN and summery network, and KL[·] is the KL divergence function. The expectation with respect to $f_{\mathbf{y}_{1:T}}(\mathbf{y}_{1:T})$ is to account for the fact that the observations are not available during the training phase. Synthetic observations of $\mathbf{y}_{1:T}$ need to be employed and the uncertainty in $\mathbf{y}_{1:T}$ needs to be considered.

According to the theory of normalizing flow given in Sect. 3.1, $\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))$ can be expressed as

$$\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T})) = f_{\mathbf{z}}(\mathbf{z} = \mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T})))\left|\det\left(\frac{\partial\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))}{\partial\boldsymbol{\theta}}\right)\right|. \tag{16}$$

Since $f_{\mathbf{z}}(\mathbf{z} = \mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))) = \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left[\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\right]^2\right\}$, we have

$$\begin{aligned}
\log\left\{\hat{f}_{\boldsymbol{\theta},\ \boldsymbol{\omega}}(\boldsymbol{\theta}|\varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\right\} = &\log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{1}{2}\left[\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))\right]^2\\
&+ \log\left|\det\left(\frac{\partial\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))}{\partial\boldsymbol{\theta}}\right)\right|.
\end{aligned} \tag{17}$$

The optimization model given in Eq. (9) can then be approximated using Monte Carlo simulation (MCS) as [31]

$$\hat{\boldsymbol{\gamma}},\ \hat{\boldsymbol{\omega}} = \mathop{\arg\min}_{\boldsymbol{\gamma},\,\boldsymbol{\omega}} \left\{ \frac{1}{N_{MCS}} \sum_{i=1}^{N_{MCS}} \left(\frac{1}{2}\left[\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta}^{(i)};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}^{(i)}|\boldsymbol{\theta}^{(i)}))\right]^2 - \log\left|\det\left(\frac{\partial \mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta};\ \varphi_{\boldsymbol{\gamma}}(\mathbf{y}_{1:T}))}{\partial\boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}^{(i)},\ \mathbf{y}_{1:T}^{(i)}}\right)\right|\right)\right\}, \tag{18}$$

where N_{MCS} is the number of MCS samples, $\boldsymbol{\theta}^{(i)}$ is the i-th MCS sample of $\boldsymbol{\theta}$, and $\mathbf{y}_{1:T}^{(i)}|\boldsymbol{\theta}^{(i)}$ is the synthetic observation generated using a forward model with inputs of $\boldsymbol{\theta}^{(i)}$.

After the model parameters $\hat{\boldsymbol{\gamma}},\ \hat{\boldsymbol{\omega}}$ are estimated, BayesFlow can be employed to efficiently obtain the posterior distribution $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}|\mathbf{y}_{1:T})$ for given $\mathbf{y}_{1:T}$. In contrast to other methods for Bayesian inference that require to repeat the entire inference procedures from scratch for variant measurement sequence, BayesFlow amortizes the inference workflow by a computationally intensive upfront training phase and a much cheaper inference phase.

In summary, BayesFlow is a fully likelihood-free Bayesian inference method, which directly approximates the posterior distribution for model updating without computing the likelihood function in the inference phase. In BayesFlow, a summary network and a cINN are trained based on synthetic observations. Summary network focuses on automatically capturing the most informative features from time series measurements and enabling for dimension reduction for model updating. The cINN is used to learn the posterior distribution of model parameters for given summary statistics. The cINN bi-directionally transforms the irregular-shape posterior distribution to a latent standard normal distribution. Based on that, the posterior samples can be obtained directly by sampling the Gaussian latent distribution and through the inverse mapping made possible by the cINN. The cINN can be considered as an inverse surrogate model that maps observations to posterior distribution directly. The advantages of BayesFlow are summarized as four-fold. First, BayesFlow is a fully likelihood-free approach that directly estimates the posterior instead of evaluating a (usually complex) likelihood function. It also strictly guarantees effective sampling process for the true posterior without any assumptions on the prior or posterior distributions. Second, BayesFlow has a favorable scalability and allows to operate very well for arbitrary measurement sequence as it amortizes the Bayesian inference. In other words, BayesFlow can reasonably deal with different sizes of datasets using a single trained model. Third, BayesFlow has a learnable summary network that is responsible for reducing data dimensionality and automatically learns the maximally informative statistics. Finally, BayesFlow is computationally efficient, especially for the problem that requires repeated parameter inference from scratch for different datasets and data sizes. In addition, the use of summary network contributes to alleviate computational burden by compressing high-dimensional data to a feasible and controllable size.

3.4 Structural damage detection using BayesFlow

As described above, BayesFlow based on cINN offers a promising solution to the challenging issues of Bayesian model updating-based damage detection discussed in Sect. 2. To apply BayesFlow to damage detection, we first parameterize the structural damage model. For a FE model as given in Eq. (1) in Sect. 2, model parameters can be represented by material and geometric properties or boundary condition. For damage detection through vibration-based mode updating, it is widely acknowledged that any structural damage leads to changes in vibration responses or modal data, is closely related to structural parameters, e.g., stiffness and mass. Damage detection is therefore usually performed by quantifying change in stiffness and mass parameters. However, in practice, only stiffness parameters are selected to be identified, since mass parameters are usually less critical. In addition, simultaneous identification of stiffness and mass parameters would result in un-identifiability issue due to the coupling of these two parameters [12]. It is also recognized that stiffness parameters are usually represented by elastic modulus rather than geometric properties, e.g., length and sectional area, since geometric properties may vary with elements and hence become uncontrollable in model updating. Instead, each structural component or structural group can be assigned to a single elastic modulus, rendering damage detection more practical and feasible [47].

In this context, the global stiffness matrix is expressed as a linear combination of sub-structural elemental stiffness matrices multiplied by updating stiffness parameters. Specifically, structural damage can be portrayed by a scalar reflecting stiffness change in each element. A general parameterization of stiffness matrix may be written as

$$\mathbf{K}_d(\boldsymbol{\theta}) = \mathbf{K}_0 + \sum_{l=1}^{N_\theta} (1 + \theta_l)\mathbf{K}_{\mathrm{ud}_l}, \tag{19}$$

in which $\mathbf{K}_{\mathrm{ud}_l}$ denotes the l-th elemental stiffness matrix under undamaged condition, $\mathbf{K}_0$ is the non-parameterized components of global stiffness matrix, $\mathbf{K}_d$ is structural global stiffness matrix under the damaged condition,N_θ is the total number of updating stiffness parameters, and θ_l denotes the l-th stiffness change parameter to be estimated corresponding to the l-th substructure, representing the relative change of stiffness from the baseline state value. For instance, $\theta_l = (E_l^d - E_l^{ud})/E_l^{ud}$, where E_l^d and E_l^{ud} are elastic moduli under damaged and baseline ("undamaged") state, respectively. It should be also noted that the choice of variation bounds of $\theta_l,\ l = 1,\ \cdots,\ N_\theta$ is a key aspect to guarantee the

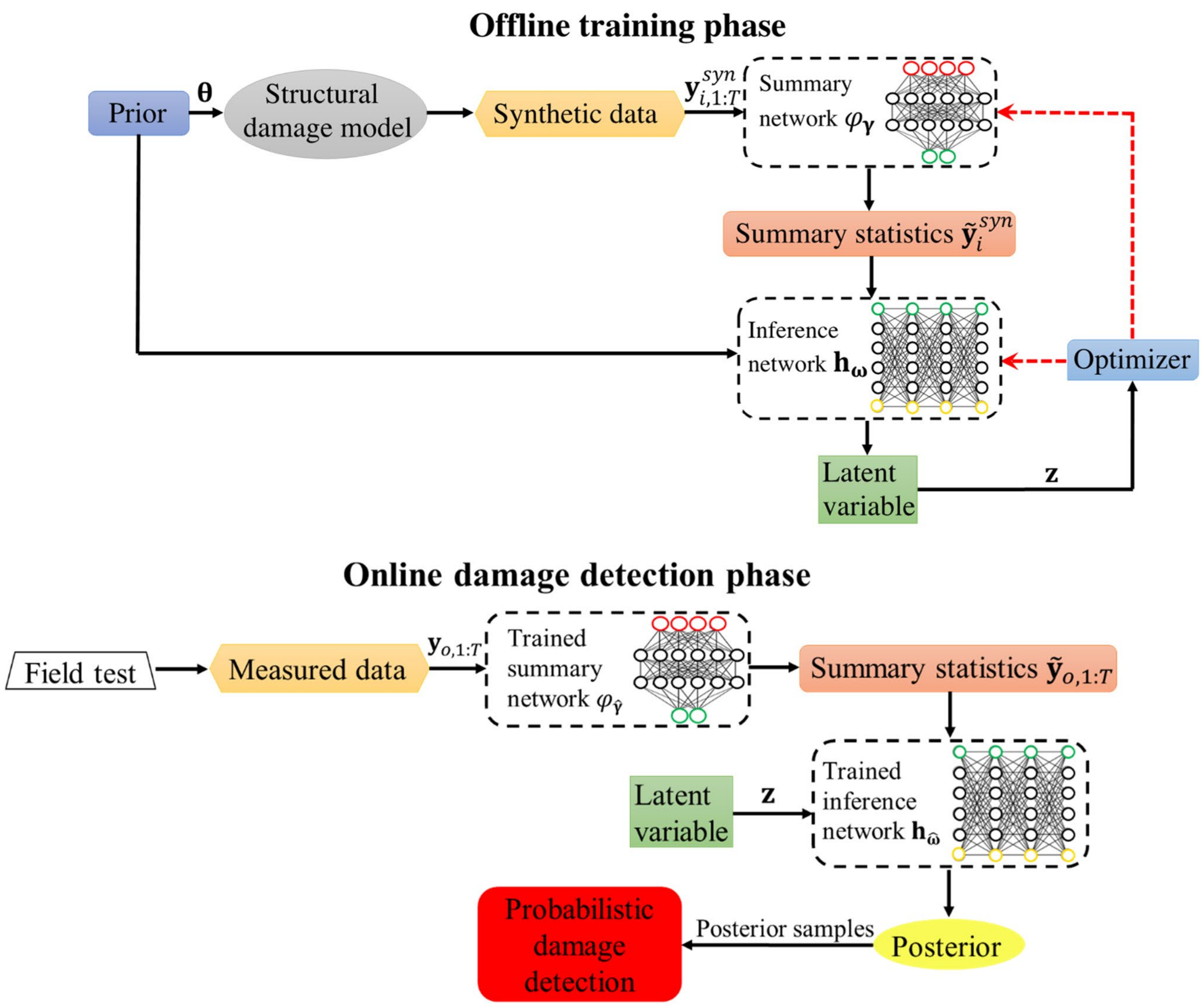

Offline training phase
Prior
$\boldsymbol{\theta}$
Structural damage model
Synthetic data
$\mathbf{y}_{i,1:T}^{syn}$
Summary network $\varphi_{\boldsymbol{\gamma}}$
Summary statistics $\tilde{\mathbf{y}}_i^{syn}$
Inference network $\mathbf{h}_{\boldsymbol{\omega}}$
Optimizer
Latent variable
$\mathbf{z}$
Online damage detection phase
Field test
Measured data
$\mathbf{y}_{o,1:T}$
Trained summary network $\varphi_{\hat{\boldsymbol{\gamma}}}$
Summary statistics $\tilde{\mathbf{y}}_{o,1:T}$
Latent variable
$\mathbf{z}$
Trained inference network $\mathbf{h}_{\hat{\boldsymbol{\omega}}}$
Posterior samples
Posterior
Probabilistic damage detection

Fig. 4 Flowchart of damage detection using BayesFlow

physical meaning. The variation bound is usually assumed based on engineering judgment. In this study, parameter bound following a uniform distribution over an interval of (− 30%, 30%) is considered according to the studies in [19, 59]. We noted that the parameterization in Eq. (19) has a limitation that the damage is homogenized over the scale of an element, which depends on how the sub-structuring is formulated. The presented method in this paper, however, is not limited to such a parameterization. It is applicable to any damage model where the damage can be parameterized.

Based on the parameterization, the task of damage detection is to estimate the posterior distribution of $\boldsymbol{\theta}$ using vibration observations $\mathbf{y}_{1:T}$ of the structure as described in Sect. 2. Figure 4 depicts the overall procedure of damage detection using BayesFlow. It consists of an offline training phase and an online detection phase. In the offline training phase, we first generate N_t training samples of $\boldsymbol{\theta}$ according to its prior distribution. Denoting the training samples as $\boldsymbol{\theta}_{train} = [\boldsymbol{\theta}_t^{(1)}, \cdots, \boldsymbol{\theta}_t^{(N_t)}]$ and by accounting for various uncertainty sources, we then obtain synthetic observation data $\mathbf{y}_{i,1:T}^{syn}$, $i = 1, \cdots, N_t$, where $\mathbf{y}_{i,1:T}^{syn}$ represents the synthetic observation generated using the i-th training sample of $\boldsymbol{\theta}$ and based on a random realization of the unmeasurable input excitation. The synthetic observations $\mathbf{y}_{i,1:T}^{syn}$, $i = 1, \cdots, N_t$ then pass to the initial summary network $\varphi_{\boldsymbol{\gamma}}(\cdot)$ to obtain the summary statistics $\tilde{\mathbf{y}}_i^{syn}$, $i = 1, \cdots, N_t$. Using the summary statistics and the initial inference network (i.e., cINN) $\mathbf{z}^{(i)} = \mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\theta}_t^{(i)}; \tilde{\mathbf{y}}_i^{syn})$, $i = 1, \cdots, N_t$, samples of latent variable $\mathbf{z}$ are obtained. The Jacobian matrix for each sample can also be computed using the cINN. After that, the objective function given in Eq. (18) can be evaluated using MCS based on the generated samples. Finally, the optimal model parameters $\hat{\boldsymbol{\gamma}}$, $\hat{\boldsymbol{\omega}}$ of the summary network and inference network (i.e., cINN) are estimated using an optimizer until the loss/objective function reaches the minimum. The obtained models can then be used in the online detection phase to enable for real-time damage detection using Bayesian model updating. It is worth mentioning that the accuracy of the Bayesian inference could be affected by the training data used trained the summary network and cINN. To ensure that the model is properly trained, we split the synthetic observation data into two parts, one part for training and the other part for validation. More training data will be added if the accuracy of validation cannot satisfy the requirement.

In the online detection phase, as illustrated in Fig, 4, measured data $\mathbf{y}_{o,1:T}$ are collected from field test, then passed through the trained summary and inference networks and obtain the posterior distributions directly without evaluating any likelihood function. The posteriors of damage parameters are approximated by samples from the latent distribution and the invertible neural network (see Sect. 3.2 generative direction transformation using cINN). Finally, the posterior samples of $\boldsymbol{\theta}$ are used for probabilistic damage detection. Note that there are many factors affecting model updating and damage detection during online monitoring, e.g., environmental change and loading conditions. However, it is quite difficult to take all factors into consideration when performing online monitoring. As the environmental and operational conditions, such as temperature, wind, traffic or other loading, etc., are nonstationary and generally uncontrollable (and sometimes unmeasurable). These unconsidered factors manifest themselves as "uncertainty" sources in Bayesian model updating. As a probabilistic damage detection method, Bayesian model updating can naturally account for various uncertainty sources. BayesFlow used in this study enables us to quantify uncertainty in damage states (i.e., posterior distribution) within just a few seconds (see Table 3). The near real-time inference using BayesFlow is a key step in enabling online health monitoring. In addition, an alarming criterion for structural damage detection is usually needed in online monitoring. However, setting a universal alarming criterion is difficult and application-specific, as different structures have their own characteristics and varied operational conditions. For instance, in some civil infrastructures, stiffness reduction exceeding 20% usually induces noticeable change in structural dynamics. An alarming threshold of 20% can then be used to inform engineers to perform necessary repairing work in that situation. Alternative to setting alarming criteria, one can detect damage by inspecting probabilistic damage curves (PDC) or cumulative distribution function (CDF) of model parameters based on uncertainty information acquired from Bayesian model updating or other stochastic model updating methods. PDCs or CDFs related to damage locations are clearly distinguishable from the ones at healthy locations [49, 60]. In other words, model parameters with outstanding and aberrant PDCs or CDFs tend to be damaged and should be of interest. Therefore, an alarming criterion may not be required for online damage detection, and the analysis of PDCs or CDFs allows to directly assess damage location and damage severity for online damage detection.

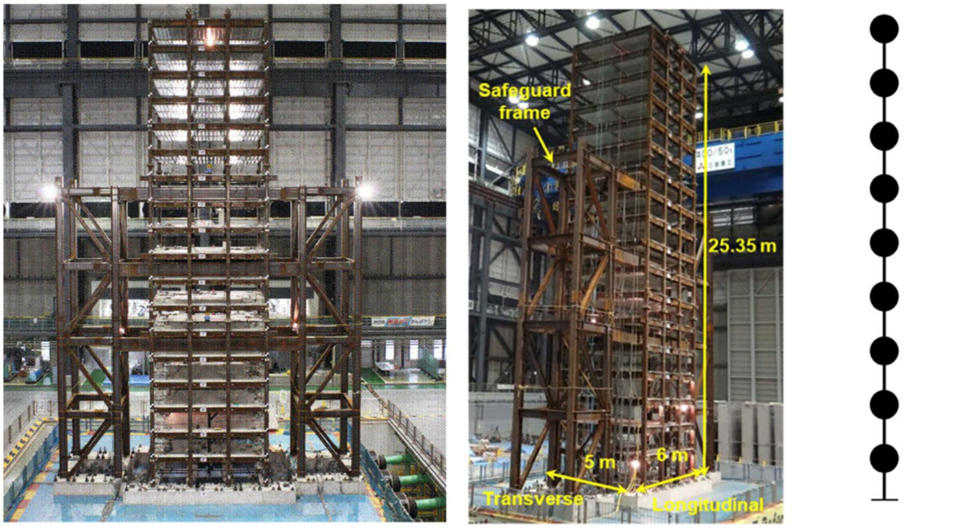
4 Case studies

In this section, BayesFlow is applied to damage detection of two benchmark examples, including an 18-story shear frame and a concrete building frame. Dynamic responses under unknown and unmeasured ambient vibration are used to identify structural parameters. BayesFlow is compared with DREAM sampling method [35] in frequency domain to verify its efficacy.

4.1 An 18-story shear frame

An 18-story shear frame is selected as the first example to validate the efficacy of damage detection using BayesFlow. A one-third scale shear frame specimen was built and tested at the E-Defense shaking table in Japan [61]. The physical structure represents the dynamic behavior of a steel high-rise building designed and constructed from 1980 to 1990s. Figure 5(a) shows the front and side view of the structure. The plane at each floor has the same dimension of 5 × 6 (meters), and the total height is 25.35 m. The total weight of the steel frame is 3500 kN excluding the foundation. This numerical study simplifies the structure as a 9-DOF shear model as shown in Fig. 5(b).

It is assumed that mass is accurately known and not included in the updated parameters. The initial stiffness for each floor is obtained from nominal material properties (i.e., elastic modulus). In this study, the stiffness change parameters representing the relative change of stiffness at each floor are selected as updating parameters, denoted as $\theta_1 \sim \theta_9$, where $\theta_i = (E^{\mathrm{act}} - E^{\mathrm{nom}})/E^{\mathrm{nom}}$, $\forall i = 1, \cdots, 9$, E^{act} and E^{nom} are respectively actual and nominal elastic moduli. θ_i, $\forall i = 1, \cdots, 9$ range from − 0.3 to 0.3 based on empirical knowledge and study in [59, 62]. Assuming that the shear frame is subjected to ambient excitation, and the excitation is unmeasured but modeled as Gaussian white noise at all floor levels with power spectral density of 3 N/$\sqrt{\mathrm{Hz}}$, similar to ambient vibration test in [63]. Only output vibration responses are measured. Due to the limited number of sensors, only incomplete data can be measured in practice. Hence the three-minutes acceleration responses are



(a) Overview of the structure (b) 9-DOF

Fig. 5 An 18-story shear frame

simulated only at the 1st, 2nd, 3rd, 5th, 7th, and 9th floor with a sampling frequency of 100 Hz based on structural dynamics and continuous state-space model implemented by functions '*ss*' and '*lsim*' in MATLAB.

4.1.1 Model training and validation

All summary and invertible networks described in Sect. 3.3 are jointly trained by calibrating network hyper-parameters. In this example, all programs are implemented in Python using TensorFlow library and a personal computer with a single CPU. The Adam Optimizer is used to minimize the KL divergence in Eq. (15) with a default learning rate of 0.001. The four subnetworks in cINN are designed as fully connected neural network with exponential linear units (ELU). The summary network is set as 1D CNN, and the cINN consists of 10 cACLs.

To generate training data, training samples of stiffness change parameter $\boldsymbol{\theta}$ are drawn from the uniform distribution $U \sim [-0.3,\ 0.3]$ using Latin hypercube sampling (LHS). Three-minute synthetic acceleration data are then simulated using an FE model of the shear frame. As a result, 800 sets of training data and additional 100 sets of test data are simulated for model training and test. During training, 30 epochs with 200 iterations per epoch are adopted. Two different metrics, coefficient of determination (R^2) and normalized root mean squared error (NRMSE), are employed to assess the accuracy of training. Figure 6 shows the validation results for the 100 sets of test data. As shown in this figure, R^2 and NRMSE of all parameters are above 0.98 and close to 0, respectively, except for θ_2 and θ_4. It implies a high agreement between the model prediction and true values. The accuracy of θ_2 and θ_4 is not as good as the others. This is probably attributed to the fact that these two parameters are not as sensitive as the others to responses.

4.1.2 Probabilistic damage detection

In this section, probabilistic damage detection is performed to identify damage location and severity using the trained BayesFlow model. One damage scenario with multiple damage locations at different floors is studied, as shown in Table 1. The initial model of this shear frame is assumed to be under healthy condition (e.g., mass and stiffness at each floor are intact). For the damaged scenario, damage severity is quantified by the percentage of stiffness change, e.g., relative change in elastic modulus. The negative sign in Table 1 denotes stiffness reduction, such as 20% and 10% stiffness reduction at the 1st and 3rd floor. Note that the stiffness reduction considered in this study is realistic and readily achieved in experimental study. For example, the stiffness reduction can be artificially created by changing geometric properties in elements, such as reducing the width of a column [64], replacing or removing structural components, e.g., braces or columns [65]. Then corresponding stiffness reduction can be calculated, which then can be used to measure the accuracy of identified stiffness reduction.

Ten sets of three-minute vibration responses at the 1st, 2nd, 3rd, 5th, 7th, and 9th floor corresponding to the damage scenario are measured. Gaussian white noise of 5% Root Mean Square (RMS) noise–signal ratio (NSR) is added to

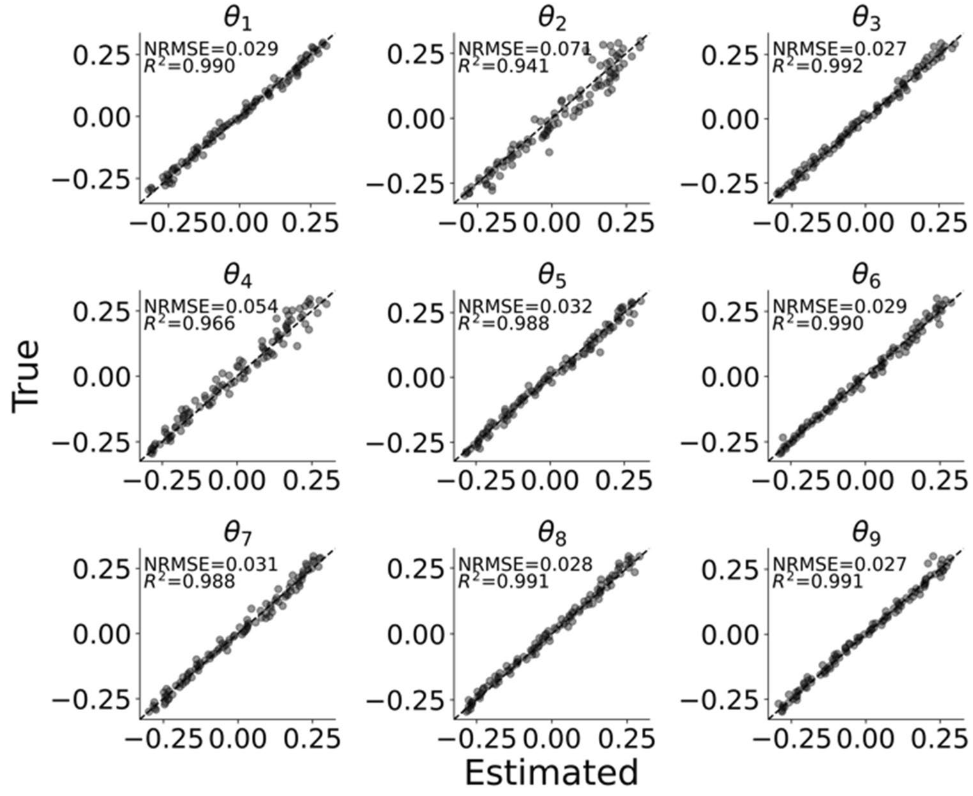
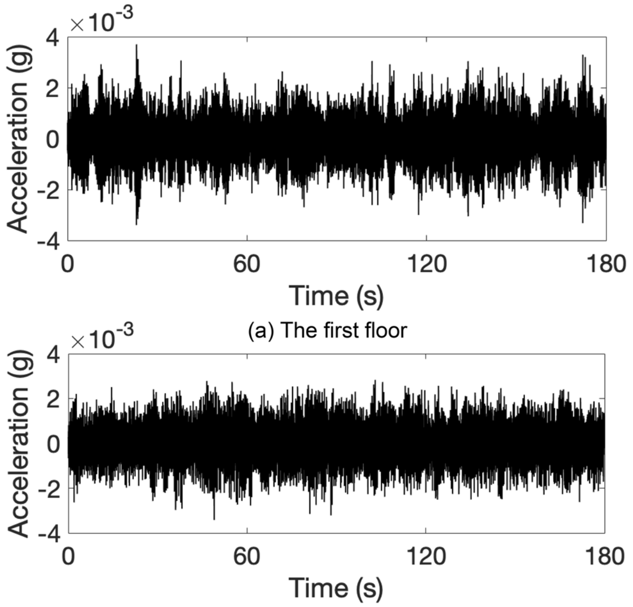


Fig. 6 Training accuracy verification of BayesFlow

Table 1 Damage location and severity of shear frame

Damage scenario	Damage location (damage severity)
1	1st (– 20%), 3rd (– 10%), 4th (– 20%), 7th (– 10%), 8th (– 20%)

all measured acceleration data to mimic additional measurement fluctuation noise. Note 5% RMS NSR is a realistic measurement noise level in real-world application [66]. The data quality with 5% noise level can be readily achieved in typical ambient vibration. Figure 7 shows examples of the measured vibration responses at the 1st and 9th floor.

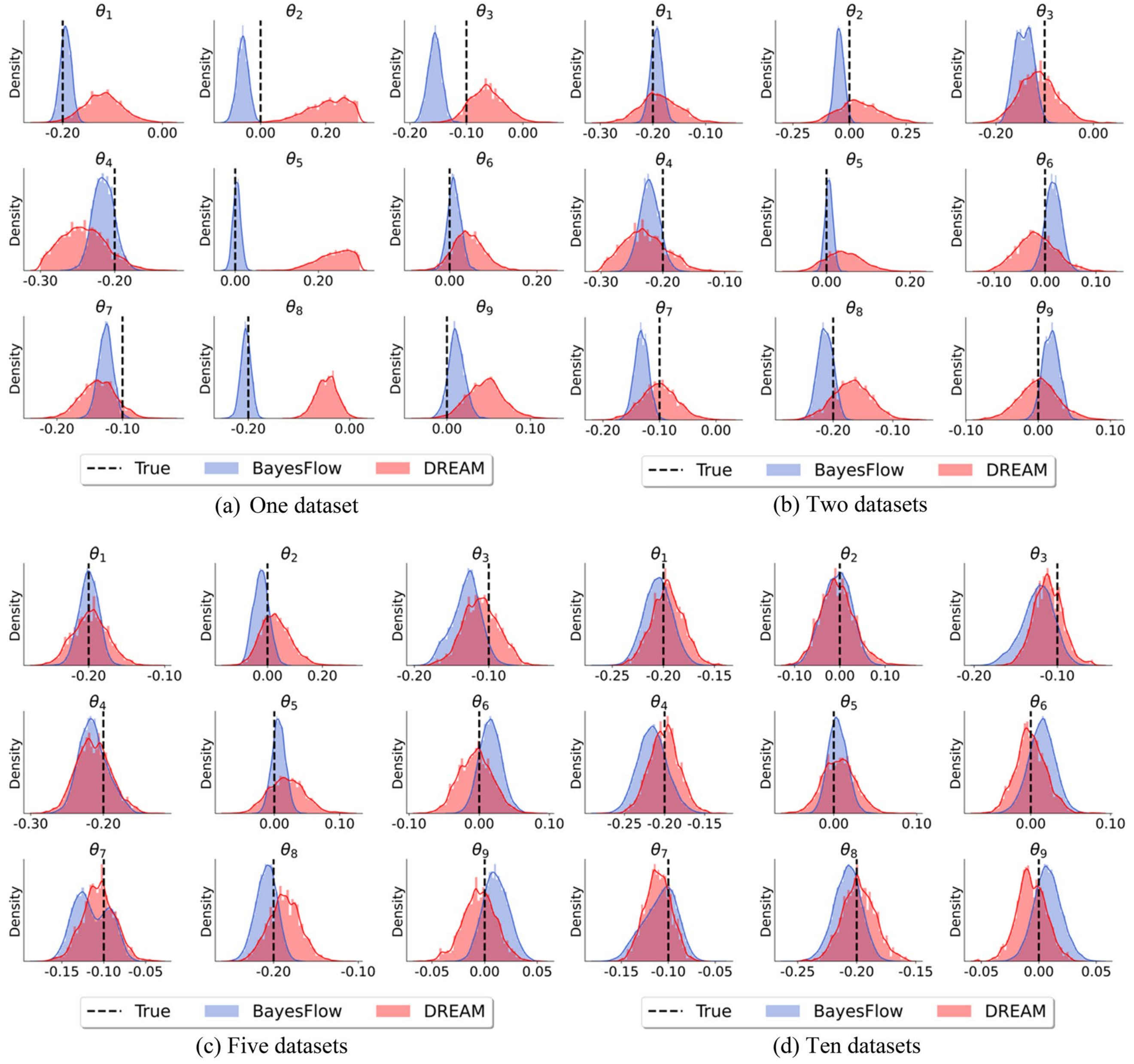


(a) The first floor

(b) The ninth floor

Fig. 7 An example of measured acceleration

As mentioned above, the performance of BayesFlow on damage detection is compared with the DREAM method. DREAM is an advanced sampling method that parallelly runs multiple Markov chains to draw samples for the target posterior [35]. When applying DREAM for damage detection, acceleration data must be transformed into frequency-domain modal data, such as natural frequencies and mode shapes, due to the assumption of ambient vibration and unmeasured excitation. Modal data can be extracted from time series data using either stochastic subspace identification [41] or Bayesian operational modal identification method [42]. For this example, the first six identified modes in damaged condition are used in DREAM. In Bayesian inference using modal data, as shown in Eqs. (5) and (6), the frequency error and mode shape error between model-derived and measured modal properties are utilized to construct likelihood function. The use of natural frequency and mode shape in DREAM has some limitations. These two



(a) One dataset

(b) Two datasets

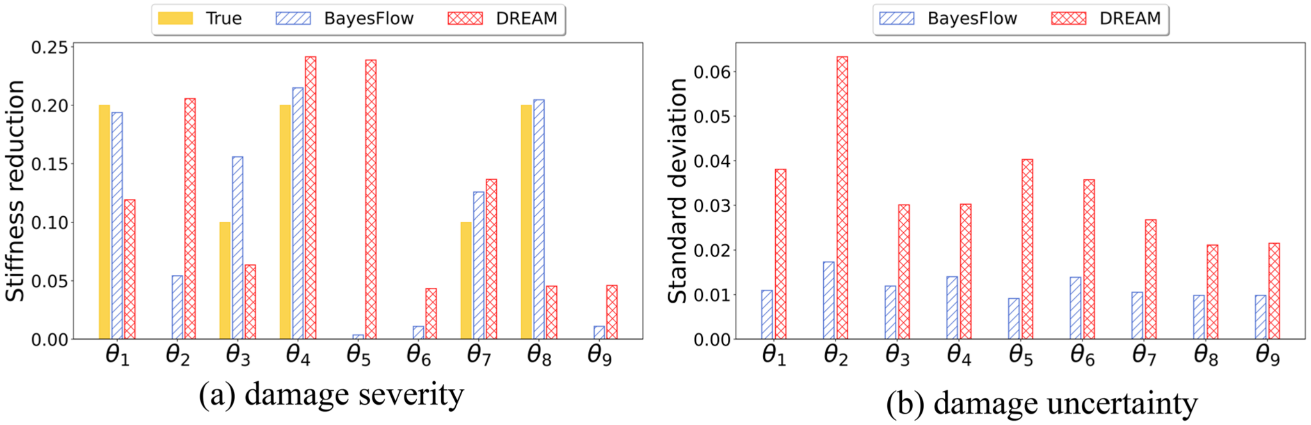
(c) Five datasets

(d) Ten datasets

Fig. 8 Posterior distributions from BayesFlow and DREAM for different numbers of datasets

features are globally identified from vibration responses. Therefore, they maybe exhibit more sensitivity to global damage, e.g., overall stiffness or mass change, but less sensitivity to local damage, e.g., a small crack or hole. One solution is to incorporate damping in Bayesian inference, since it is recognized that damping is more sensitive to local change, such as structural internal change, than natural frequency and mode shape [12, 45]. In this study, only global damage, such as overall stiffness reduction, is considered to compare DREAM with the proposed method. It is worth noting that acceleration data in time domain are directly used in the proposed method using BayesFlow without transforming the data into frequency-domain modal data. The summary network in BayesFlow automatically extracts important features for model updating. This is one of the advantages of BayesFlow over the conventional approaches.

Figure 8 presents the comparison of posterior distributions of damage parameters obtained by BayesFlow and DREAM using different number of datasets. The results show that BayesFlow has a more stable performance for damage detection than DREAM when the number of datasets varies. For example, when only a single set of acceleration data is utilized, DREAM has a very poor performance for damage detection (see Fig. 8a). The posterior mean obtained

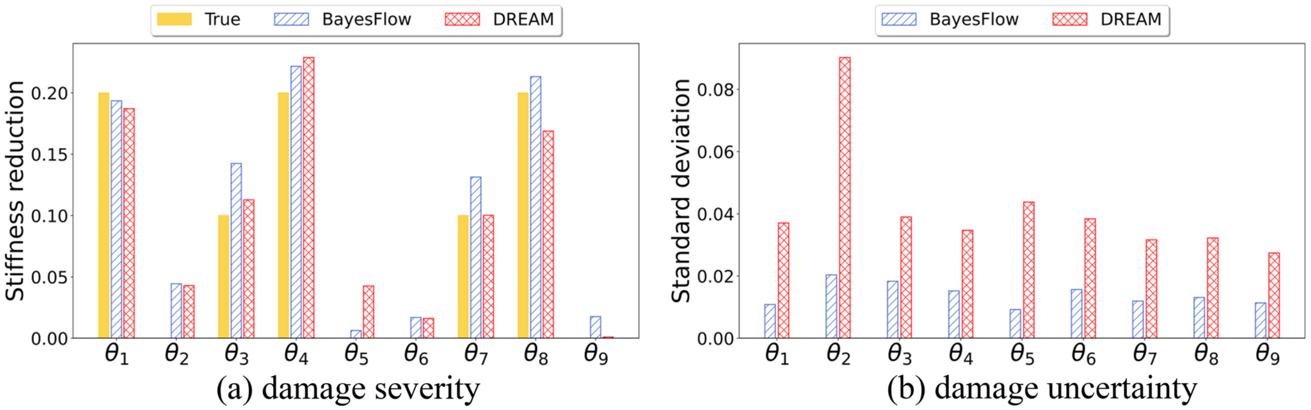


(a) damage severity

(b) damage uncertainty

Fig. 9 Damage identification by one datasets on shear frame

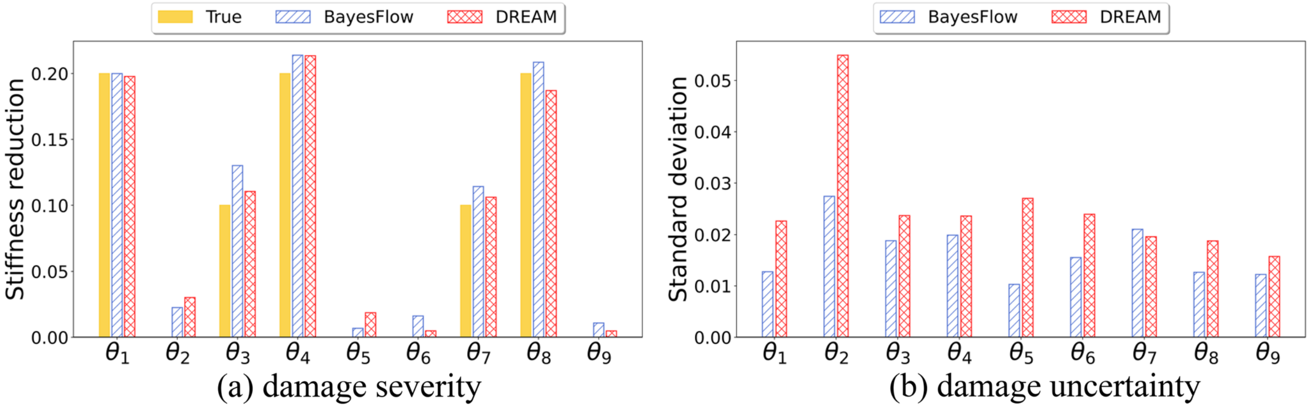
from DREMA deviates from the ground truth, especially for θ_2, θ_5, and θ_8, suggesting a failure of damage detection by DREAM. However, BayesFlow can still accurately captures the damage using one set of acceleration data. As the number of datasets increases, the performance of DREAM gets better and closer to that of BayesFlow. The estimated posterior distributions overlap with each other in Fig. 8(d), implying



(a) damage severity

(b) damage uncertainty

Fig. 10 Damage identification by two datasets on shear frame



(a) damage severity

(b) damage uncertainty

Fig. 11 Damage identification by five datasets on shear frame

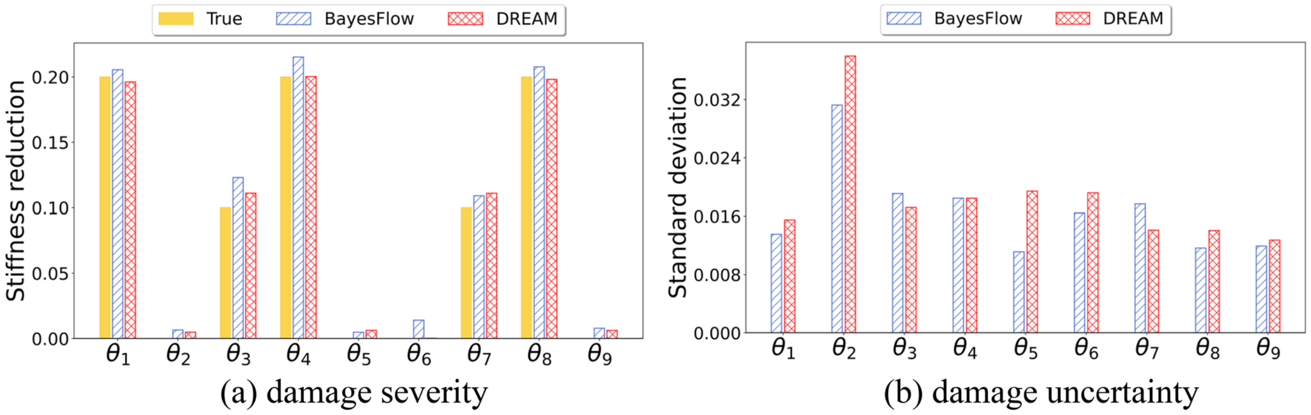


Fig. 12 Damage identification by ten datasets on shear frame

a good agreement between BayesFlow and DREAM using 10 datasets. But it is observed that some parameters (e.g., θ_5) identified by DREAM have larger uncertainty than their counterparts obtained from BayesFlow.

Figures 9, 10, 11 and 12 show the comparison of posterior mean and standard deviation of the identified stiffness reduction by BayesFlow and DREAM. As shown in Fig. 9, DREAM gives many false alarms of damage detection and large uncertainty when only one dataset is available. When the number of available datasets increases (i.e., from Figs. 9 to 12), the accuracy of identified damage severity by DREAM is greatly improved and becomes similar to the accuracy achieved by BaysFlow. The results also show that the standard deviations of posterior distributions from BayesFlow are overall smaller than that from DREAM, indicating a more reliable damage detection using BayesFlow.

As discussed in Sect. 3.3, one of appealing features of BayesFlow is its ability to perform parameter inference for different sizes of measurements using only one trained model. To demonstrate this capability, six cases with different data durations (ranging from 0.5 min to 3 min with a step of 0.5 min) are considered for damage detection using BayesFlow. Figure 13 presents the damage identification results with respect to different data durations. The black dot denotes the posterior mean, blue shaded area denotes 95% confidence interval (CI), and red dashed line denotes the ground truth. The results indicate that the posterior means of all parameters overall tend to be more accurate when the time duration for data collection becomes longer. In addition, there is persistent uncertainty in the posterior distributions. This may be attributed to the fact that for structural damage detection under ambient vibration, only vibration responses are measured, and the excitation is unknown. The results in Fig. 13 show that BayesFlow can successfully identify structural damage using arbitrary sizes of dataset using just one trained model. There is no need to build another model from scratch. This is a practical and convenient property, particularly for continuous SHM involving plenty of data analysis and different data information.

4.2 A concrete building frame

A concrete building frame, representing a full-scale test structure in Structural Engineering and Materials Laboratory on Georgia Tech Campus [67], is employed as a second example. The building frame targets on investigating the structural behavior of typical low-rise reinforced concrete office buildings in the central and eastern United States built from 1950 to 1970s. The structure constitutes four identical frames (numbered as #1–#4) and two collapse frames. All frames are distributed separately with each other with a gap between every two adjacent ones, so that each frame can be modeled and analyzed independently. In this study, the BayesFlow is used to perform structural damage detection for frame #1. Figure 14(a) shows the front, elevation, and side view of frame #1.

The columns and beams of frame #1 are modeled by frame elements in SAP2000, as depicted in Fig. 14(b). In SAP2000, the entire structure is simplified as an FE model with 2302 DOFs. The mass matrix is a diagonal matrix with zero element at rotational direction. A more detailed FE model information can be found in [68]. In concrete building, a total of six stiffness change parameters are considered to be updated in this example, denoted as $\theta_1 \sim \theta_6$, as shown in Fig. 14(b), where $\theta_1 \sim \theta_4$ respectively represents the

Fig. 13 Damage identification using BayesFlow with varied data size

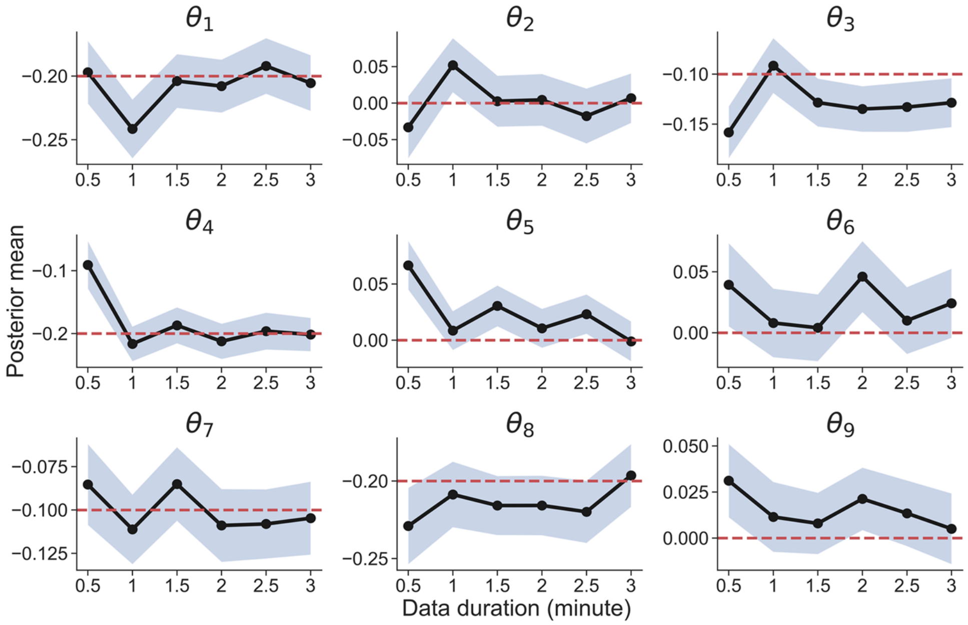
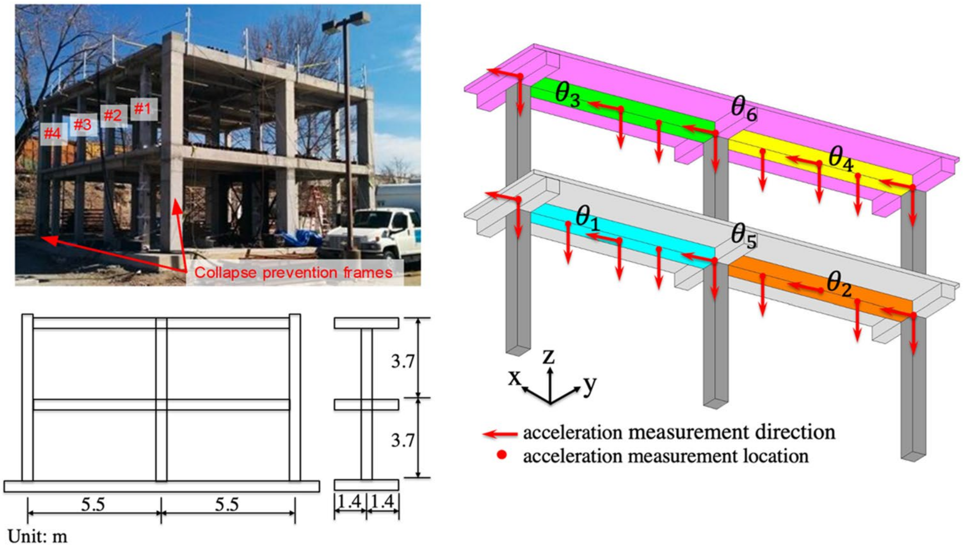


Fig. 14 Concrete building frame



(a) View of the structure (b) Model of frame #1

relative change between nominal and actual elastic modulus of longitudinal beam members (x direction) at the first and second floor. $\theta_5 \sim \theta_6$ respectively represents the relative change in elastic modulus of the first and second slab and associated lateral beam members (y direction). While material properties in columns are assumed to be known accurately and thus not updated here.

Dynamic vibration test is simulated to measure accelerations under ambient vibration. Accelerometers shown in Fig. 14(b) are deployed at two slabs to measure vertical and longitudinal vibrations (z and x directions). Only a total of 26 DOFs are measured to mimic the reality. Four-minute acceleration responses are measured with a sampling frequency of 100 Hz. Similar to the previous example, the excitation is also modeled as Gaussian White noise with power spectral density of $3\,\mathrm{N}/\sqrt{\mathrm{Hz}}$.

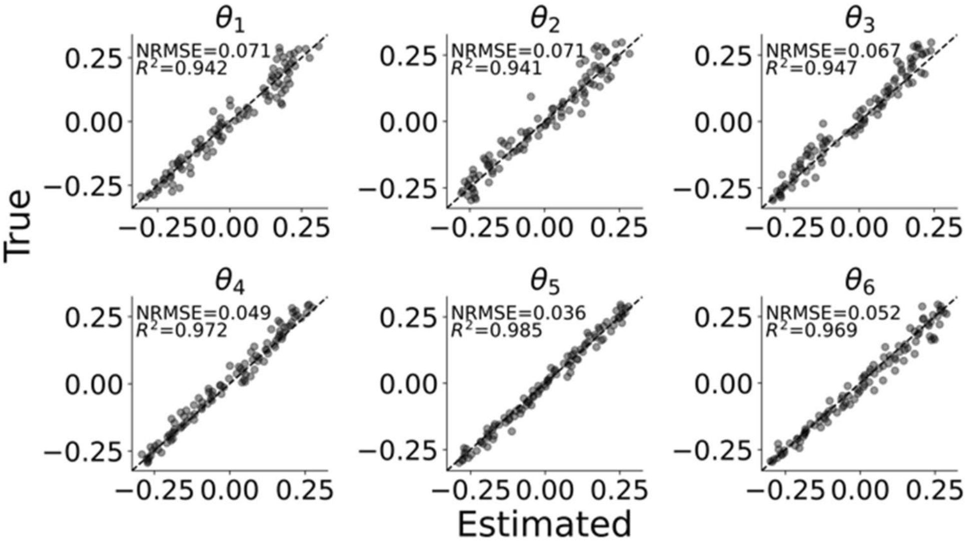


Fig. 15 Training accuracy verification of BayesFlow

Table 2 Damage location and severity of concrete building

Damage scenario	Damage location (damage severity)
1	θ_1(– 10%), θ_3 (– 20%), θ_4 (– 10%), θ_5 (– 20%)

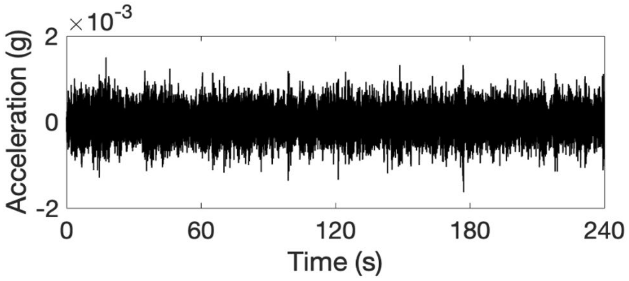


Fig. 16 An example of measured acceleration for 4 min

4.2.1 Model training and validation

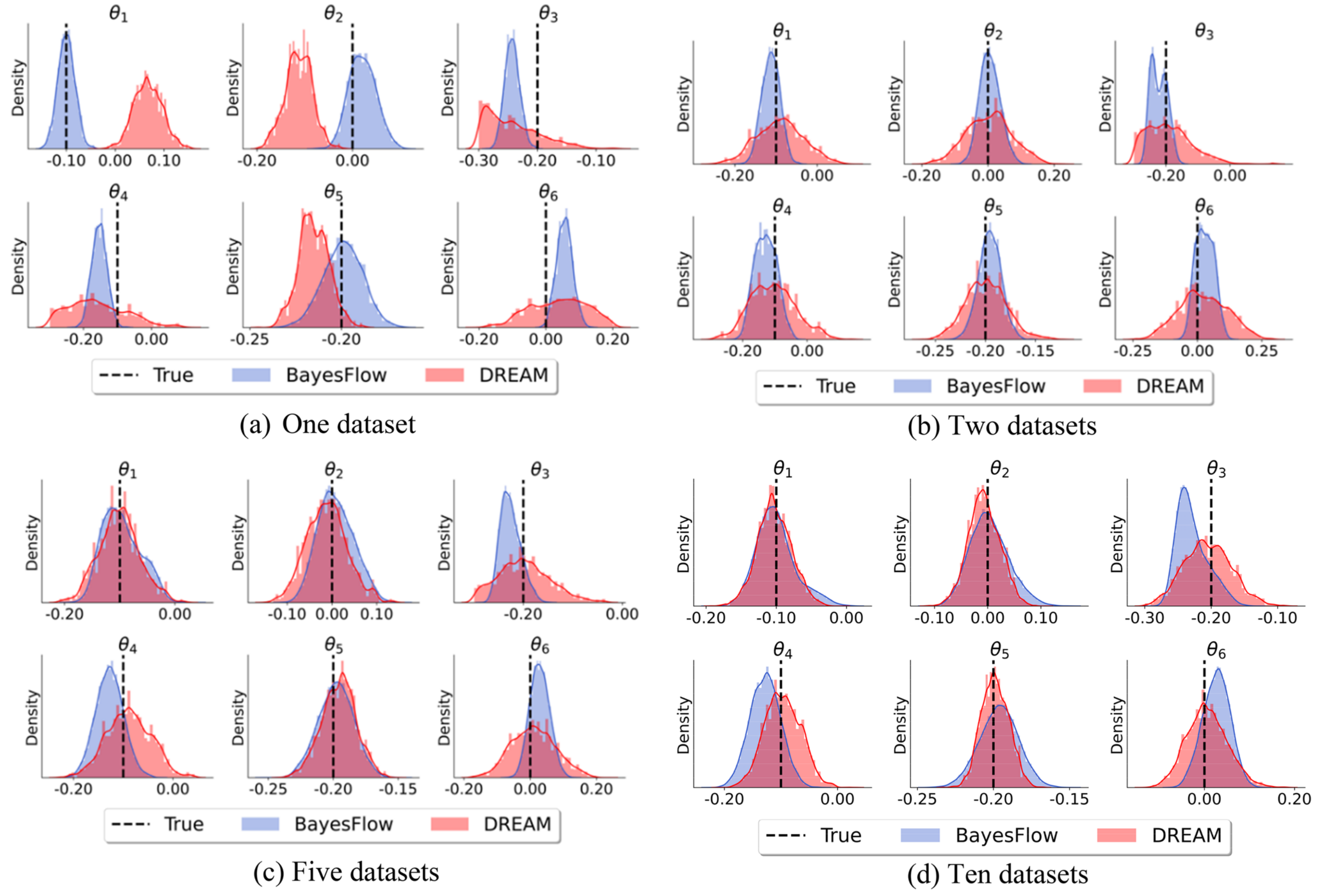
The same initial setting of summary and invertible networks as the first example are used in BayesFlow for this example. 800 sets of training samples in terms of six parameters are generated from uniform distribution $U \sim [-0.3, 0.3]$ using LHS, resulting in 800 sets of synthetic acceleration responses simulated from the FE model. Also similar to the first example, an extra 100 sets of data are generated to verify the accuracy of the trained model. 40 epochs with 200 iterations each are adopted for training all neural networks. Figure 15 presents the validation results. As shown in this figure, R^2 and NRMSE for all parameters exceed 0.94 and approach to 0 respectively, indicating accurate prediction by BayesFlow. It is also observed that $\theta_4 \sim \theta_6$ appear to be well recovered, but $\theta_1 \sim \theta_3$ turn out to be more difficult to estimate. This is probably because of $\theta_1 \sim \theta_3$ are less sensitive to vibration responses compared to $\theta_4 \sim \theta_6$. In addition, the concrete building is more complex compared to shear frame as given in example 1 (2302 DOFs vs 9 DOFs), which may increase the difficulty in parameter estimation with limited measurements.

4.2.2 Probabilistic damage detection

Commensurate with example 1, one damage scenario with multiple damage locations is artificially introduced to demonstrate BayesFlow's capability of detecting damage. Table 2 lists the assumed damage location and severity. The structural damage is defined as stiffness reduction represented as the relative change of elastic moduli of beams and slabs, which is similar to that in example 1.

Ten sets of vibration responses at 26 DOFs corresponding to the damage scenario are measured. Gaussian white noise of 5% NSR is added to the measured accelerations again. Figure 16 shows an example of one measured acceleration response.

DREAM is also applied for damage detection using the same measurements, in which 20,000 samples are generated to estimate the posteriors. Same as the previous example, modal data is identified from accelerations when applying the DREAM, containing the first eight natural frequencies and mode shapes, since the excitation is unmeasured in this study. Figure 17 gives the results of the posterior estimates by BayesFlow and DREAM. Vey similar observations as that from the shear frame example, BayesFlow can detect structural damage across different datasets. The performance of DREAM, however, is significantly affected by the number of datasets. For instance, for one dataset case, it can be seen in Fig. 17(a) that the posterior estimates from DREMA either



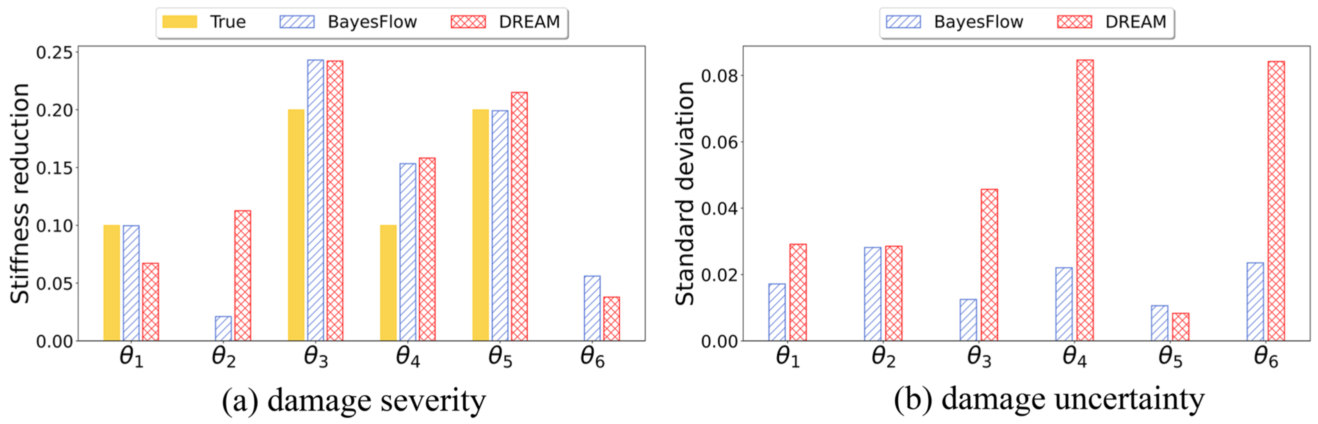
(a) One dataset

(b) Two datasets

(c) Five datasets

(d) Ten datasets

Fig. 17 Posterior distributions from BayesFlow and DREAM for different numbers of datasets



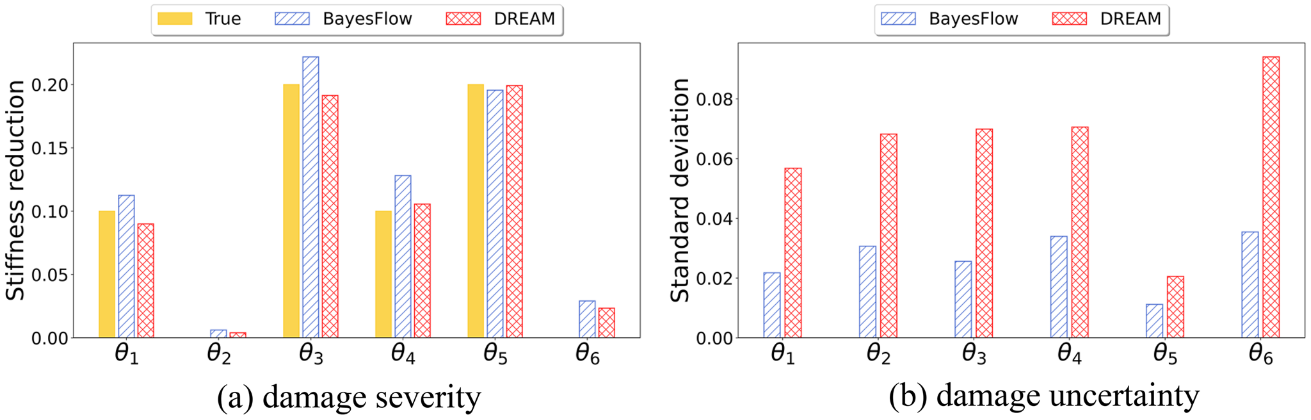
(a) damage severity

(b) damage uncertainty

Fig. 18 Damage identification by one datasets on concrete building

deviate from the true values (e.g., $\theta_1 \sim \theta_3$) or have very large uncertainty (e.g., $\theta_3,\ \theta_4,\ \theta_6$). As in the first example, when more and more data available, the approximate posteriors from two methods get closer and closer to each other.

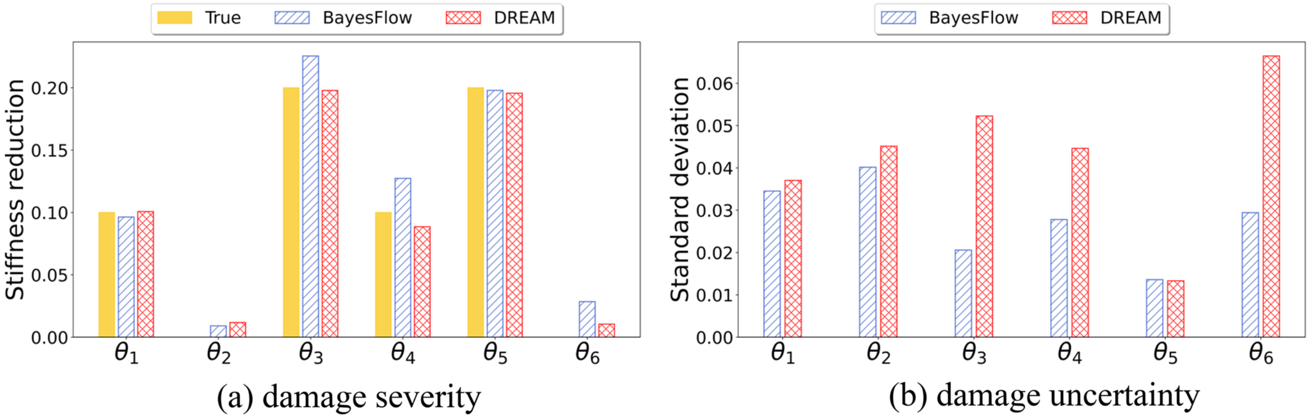
Figures 18, 19, 20 and 21 present the results of posterior mean and standard deviation obtained from BayesFlow and DREAM for different time durations of measurements. Overall, BayesFlow outperforms DREAM, which



(a) damage severity

(b) damage uncertainty

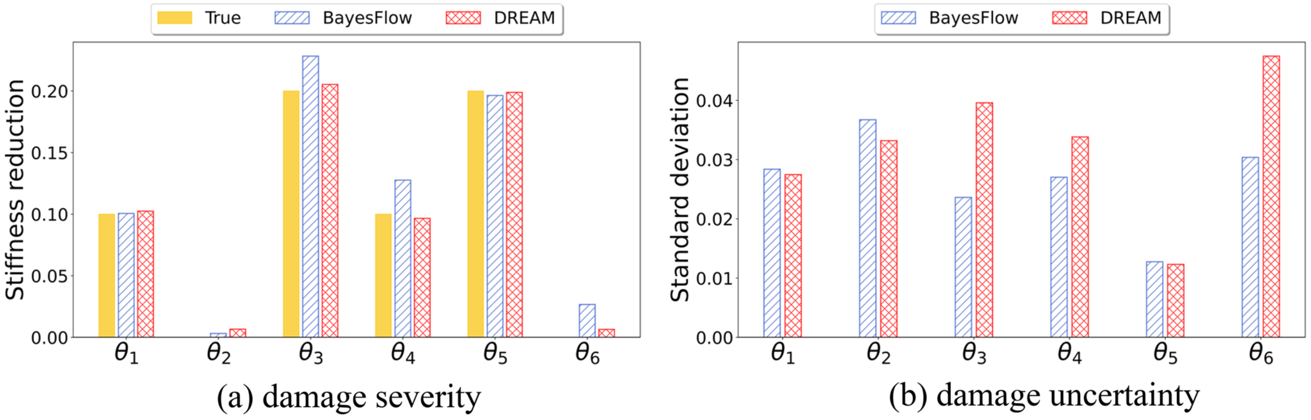
Fig. 19 Damage identification by two datasets on concrete building



(a) damage severity

(b) damage uncertainty

Fig. 20 Damage identification by five datasets on concrete building



(a) damage severity

(b) damage uncertainty

Fig. 21 Damage identification by ten datasets on concrete building

is consistent with the results in Figs. 9, 10, 11 and 12 in Sect. 4.1.2. As shown in Fig. 18, when only one dataset is available, BayesFlow accurately identifies stiffness reduction while DREAM falsely detects the damage severity for θ_2. In addition, although the stiffness reduction for $\theta_3 - \theta_5$ identified by two methods are similar, the uncertainty of posterior distributions from DREAM is much higher than

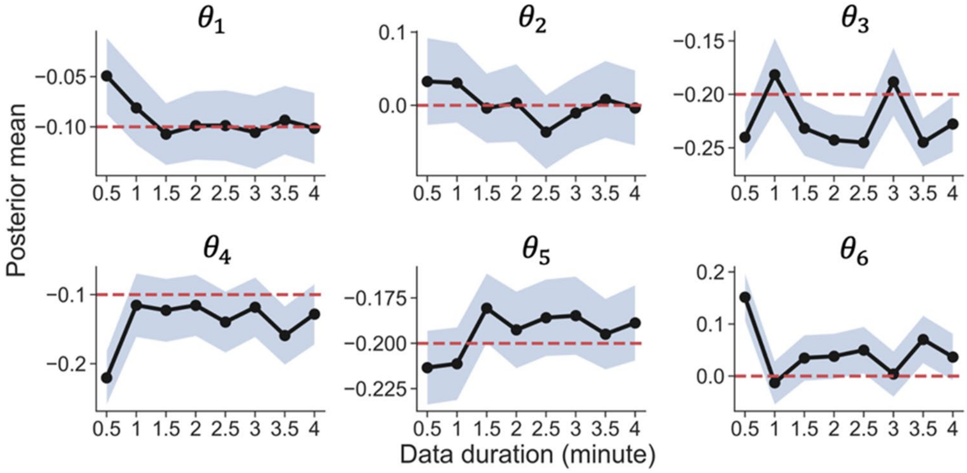


Fig. 22 Damage identification across varied data size on concrete building

Table 3 Comparison of computational cost between BayesFlow and DREAM

Cases	BayesFlow		DREAM
	Training (h)	Inference (s)	Inference (h)
18-story shear frame	23	**5**	1.2
Concrete building frame	25	**7**	3.2

The bold values just emphasize the computational cost by BayesFlow

that from BayesFlow. These results again demonstrate that BayesFlow performs better than DREAM for probabilistic damage detection from two aspects: (1) BayesFlow has a stable and robust performance on damage detection given different amount of measurement data; (2) BayesFlow identifies damage severity with less uncertainty, indicating higher confidence on damage detection.

As analogous to previous example, eight cases with different data duration are considered for the purpose of damage detection using BayesFlow. Measurement duration ranges from 0.5 to 4 min with a step of 0.5 min. Figure 22 shows the posterior results of different damage parameters. It clearly shows that when the time duration for data collection gets longer, the identified posterior mean overall tends to be more accurate and converges to the true values. The ability of working with different sizes of dataset indicates that BayesFlow has great potential in performing long-term SHM with varied number of observations due to restricted conditions of data acquisition.

4.3 Summary of computational time

Table 3 summaries the computational cost of BayesFlow and DREAM for damage detection using ten data sets in the two examples. It is noted that BayesFlow takes around 23 h and 25 h for training for shear frame and concrete building, respectively. After training, it takes less than 10 s to perform damage detection on ten datasets. On the contrary, DREAM takes about 1.2 h and 3.2 h respectively to complete the task of damage detection. Despite the substantially higher time required for training using BayesFlow, one can opt to perform offline training using simulated data. Subsequently, real-time damage detection can be realized within a few seconds using measured data from the field. Furthermore, the studied structures in this work, e.g., an 18-story shear frame modeled with 9 DOFs and a concrete building frame modeled with 2302 DOFs, is relatively simpler compared to real-world engineering structures that are usually complex and large-scale. Such structures are often modeled as high-fidelity FE models consisting of hundreds of thousands of elements and nodes. Therefore, it would take a few minutes to run such models once. To ensure a satisfactory convergence, it usually requires a huge amount of model evaluations, e.g., at least 10^4 times. In the case of limited computational budget, performing DREAM or other sampling-based methods for SHM is impractical due to the required prohibitive computational cost. BayesFlow provides a promising alternative for real-time online model updating and damage detection. Once a pre-trained model is obtained offline without the disturbance of field test, the damage detection can be efficiently conducted online within a few seconds, while DREAM or other conventional methods such as ABC may take hours to perform one model updating.

5 Conclusion

In this paper, we have applied a novel adaptation of a new likelihood-free Bayesian inference method named BayesFlow to probabilistic damage detection for an SHM application. The benefits of exploring BayesFlow in the context of structural damage detection are multifold. First, in many cases, the likelihood function is analytical intractable and not available in close from due to model complexity. BayesFlow is fully likelihood-free, which directly approximates the posterior without evaluating the likelihood function. Second, BayesFlow introduces a summary network that automatically learns the maximal information from data, rather than hand-crafted features. The raw data are compressed into a fixed-length vector, which alleviates the computational burden. Third, BayesFlow is computational very efficient for online damage detection. It allows for amortized inference. Although the required computational cost is high for training which can be conducted offline, the trained networks can efficiently estimate the posterior online given any measurements within just a few seconds. Although BayesFlow was recently developed by Radev et al. [31] in 2020. To date, likelihood-free Bayesian inference using cINN has not been explored in SHM field, especially for probabilistic damage detection and model updating. The main contribution of this work is that it is the first attempt to investigate the capability of BayesFlow (i.e., a summary network and a cINN) on structural damage detection. A new likelihood-free Bayesian inference is introduced to the engineering community of SHM, which would provide new insights on structural damage detection and deliver a new solution for online monitoring.

The developed method is applied to two benchmark examples, including an 18-story shear frame and a more challenging concrete building frame. Synthetic acceleration data are simulated from FE models under ambient vibration and then used to train all networks in BayesFlow. The pretrained model then efficiently performs parameter estimation given new data under damaged condition. Throughout all examples, BayesFlow exhibits superior accuracy and reliability in damage detection compared to a sampling-based method called DREAM. BayesFlow can directly work on time series data even if excitation is unmeasured, but modal data extracted from time series data have to be employed in DREAM under ambient vibration. In summary, the performance of BayesFlow is stable and robust on different amount of datasets for damage detection. The uncertainty of damage detection from BayesFlow is overall lower than that from DREAM. Furthermore, BayesFlow can perform damage detection with varied data sizes using only one trained model, which is a major advantage in practice, particularly for long-term SHM. BayesFlow also has a much cheaper inference work compared to DREAM for the two examples. Although BayesFlow takes a long time in the training phase, the training can be carried out offline, then damage detection would become real time given new data collected from field.

It is worth noting that BayesFlow is not limited to the structures studied in this paper, as it can be extended to other types of complex structures exhibiting high nonlinearity. The cINN architecture enables for non-linear bijective transformation, and there are no assumptions on model types and posteriors. The extension of BayesFlow to more complex structures and different purposes of SHM, e.g., damage prognosis and reliability analysis, will be further studied in our future work.

Acknowledgements Funding for this work was provided by the US Army Corps of Engineers through the U.S. Army Engineer Research and Development Center Research Cooperative Agreement W912HZ-17-2-0024. The support is gratefully acknowledged.

References

1. Ou J, Li H (2010) Structural health monitoring in mainland China: review and future trends. Struct Health Monit 9(3):219–231
2. Morgan CJ, Sparling BF, Wegner LD (2022) Use of structural health monitoring to extend the service life of the Diefenbaker Bridge. J Civ Struct Heal Monit 12(4):913–929
3. Song M, Yousefianmoghadam S, Mohammadi M-E et al (2018) An application of finite element model updating for damage assessment of a two-story reinforced concrete building and comparison with lidar. Struct Health Monit 17(5):1129–1150
4. Baggio C, Sabbatini V, Santini S et al (2021) Comparison of different finite element model updates based on experimental onsite testing: the case study of San Giovanni in Macerata. J Civ Struct Heal Monit 11(3):767–790
5. Zhang Z, Sun C (2021) Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. Struct Health Monit 20(4):1675–1688
6. Das S, Saha P (2018) Structural health monitoring techniques implemented on IASC–ASCE benchmark problem: a review. J Civ Struct Heal Monit 8(4):689–718
7. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. Computational intelligence for multimedia big data on the cloud with engineering applications. p. 185–231.
8. Hashim FA, Hussain K, Houssein EH et al (2021) Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. Appl Intell 51(3):1531–1551
9. Crespo LG, Kenny SP, Giesy DP (2014) The NASA Langley multidisciplinary uncertainty quantification challenge. In: 16th AIAA non-deterministic approaches conference
10. Mares C, Mottershead J, Friswell M (2006) Stochastic model updating: part 1—theory and simulated example. Mech Syst Signal Process 20(7):1674–1695
11. Jesus A, Brommer P, Westgate R et al (2019) Modular Bayesian damage detection for complex civil infrastructure. J Civ Struct Heal Monit 9(2):201–215

12. Zeng J, Kim YH (2020) Identification of structural stiffness and mass using Bayesian model updating approach with known added mass: numerical investigation. Int J Struct Stab Dyn 20(11):2050123
13. Ni P, Li J, Hao H et al (2021) Probabilistic model updating via variational Bayesian inference and adaptive Gaussian process modeling. Comput Methods Appl Mech Eng 383:113915
14. Baisthakur S, Chakraborty A (2021) Experimental verification for load rating of steel truss bridge using an improved Hamiltonian Monte Carlo-based Bayesian model updating. J Civ Struct Heal Monit 11(4):1093–1112
15. Beck JL, Katafygiotis LS (1998) Updating models and their uncertainties. I: Bayesian statistical framework. J Eng Mech 124(4):455–461
16. Durbin J, Koopman SJ (2012) Time series analysis by state space methods, vol 38. OUP Oxford, Oxford
17. Huang Y, Shao C, Wu B et al (2019) State-of-the-art review on Bayesian inference in structural system identification and damage assessment. Adv Struct Eng 22(6):1329–1351
18. Van Damme CI, Allen MS, Hollkamp JJ (2020) Updating geometrically nonlinear reduced-order models using nonlinear modes and harmonic balance. AIAAJ 58(8):3553–3568
19. Wan H-P, Ren W-X (2016) Stochastic model updating utilizing Bayesian approach and Gaussian process model. Mech Syst Signal Process 70–71:245–268
20. Adhikari S, Khodaparast HH (2014) A spectral approach for fuzzy uncertainty propagation in finite element analysis. Fuzzy Sets Syst 243:1–24
21. Sung H, Chang S, Cho M (2020) Reduction method based structural model updating method via neural networks. In: AIAA Scitech 2020 Forum
22. Trehan S, Carlberg KT, Durlofsky LJ (2017) Error modeling for surrogates of dynamical systems using machine learning. Int J Numer Meth Eng 112(12):1801–1827
23. Wan H-P, Ren W-X (2015) A residual-based Gaussian process model framework for finite element model updating. Comput Struct 156:149–159
24. Sunnåker M, Busetto AG, Numminen E et al (2013) Approximate bayesian computation. PLoS Comput Biol 9(1):e1002803
25. Fang S-E, Chen S, Lin Y-Q et al (2019) Probabilistic damage identification incorporating approximate Bayesian computation with stochastic response surface. Mech Syst Signal Process 128:229–243
26. Fernández J, Chiachío M, Chiachío J et al (2022) Uncertainty quantification in neural networks by approximate Bayesian computation: application to fatigue in composite materials. Eng Appl Artif Intell 107:104511
27. Ritto TG, Beregi S, Barton DAW (2022) Reinforcement learning and approximate Bayesian computation for model selection and parameter calibration applied to a nonlinear dynamical system. Mech Syst Signal Process 181:109485
28. Kitahara M, Bi S, Broggi M et al (2022) Nonparametric Bayesian stochastic model updating with hybrid uncertainties. Mech Syst Signal Process 163:108195
29. Barros J, Chiachío M, Chiachío J et al (2022) Adaptive approximate Bayesian computation by subset simulation for structural model calibration. Comput-Aided Civil Infrastruct Eng 37(6):726–745
30. Fang SE, Chen S (2022) A grey Bayesian inference framework for structural damage assessment. Struct Control Health Monitor 29(3):e2889
31. Radev ST, Mertens UK, Voss A, et al (2020) BayesFlow: learning complex stochastic models with invertible neural networks. IEEE Transactions on Neural Networks and Learning Systems
32. Cranmer K, Brehmer J, Louppe G (2020) The frontier of simulation-based inference. Proc Natl Acad Sci 117(48):30055–30062
33. Chib S, Jeliazkov I (2001) Marginal likelihood from the Metropolis-Hastings output. J Am Stat Assoc 96(453):270–281
34. Haario H, Laine M, Mira A et al (2006) DRAM: efficient adaptive MCMC. Stat Comput 16(4):339–354
35. Vrugt JA (2016) Markov chain Monte Carlo simulation using the DREAM software package: theory, concepts, and MATLAB implementation. Environ Model Softw 75:273–316
36. Liu JS, Chen R (1998) Sequential Monte Carlo methods for dynamic systems. J Am Stat Assoc 93(443):1032–1044
37. Zhang F-L, Au S-K (2016) Fundamental two-stage formulation for Bayesian system identification, part II: application to ambient vibration data. Mech Syst Signal Process 66:43–61
38. Wenzel H, Pichler D (2005) Ambient vibration monitoring. John Wiley & Sons, New York
39. Brownjohn JM, De Stefano A, Xu Y-L et al (2011) Vibration-based monitoring of civil infrastructure: challenges and successes. J Civ Struct Heal Monit 1(3):79–95
40. Juang J-N, Pappa RS (1985) An eigensystem realization algorithm for modal parameter identification and model reduction. J Guid Control Dyn 8(5):620–627
41. Zeng J, Kim YH (2021) A two-stage framework for automated operational modal identification. Struct Infrastruct Eng 1–20
42. Au S-K (2011) Fast Bayesian FFT method for ambient modal identification with separated modes. J Eng Mech 137(3):214–226
43. Behmanesh I, Moaveni B (2015) Probabilistic identification of simulated damage on the Dowling Hall footbridge through Bayesian finite element model updating. Struct Control Health Monit 22(3):463–483
44. Luo J, Huang M, Xiang C et al (2022) Bayesian damage identification based on autoregressive model and MH-PSO hybrid MCMC sampling method. J Civ Struct Heal Monit 12(2):361–390
45. Mustafa S, Matsumoto Y (2017) Bayesian model updating and its limitations for detecting local damage of an existing truss bridge. J Bridge Eng 22(7):04017019
46. Ding Z, Li J, Hao H (2019) Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference. Mech Syst Signal Process 132:211–231
47. Yang J-H, Lam H-F (2018) An efficient adaptive sequential Monte Carlo method for Bayesian model updating and damage detection. Struct Control Health Monitor 25(12):e2260
48. Zhou X, Kim C-W, Zhang F-L et al (2022) Vibration-based Bayesian model updating of an actual steel truss bridge subjected to incremental damage. Eng Struct 260:114226
49. Zeng J, Kim YH (2022) Probabilistic damage detection and identification of coupled structural parameters using Bayesian model updating with added mass. J Sound Vib 539:117275
50. Das S, Saha P, Patro SK (2016) Vibration-based damage detection techniques used for health monitoring of structures: a review. J Civ Struct Heal Monit 6(3):477–507
51. Feng Z, Lin Y, Wang W et al (2020) Probabilistic updating of structural models for damage assessment using approximate Bayesian computation. Sensors 20(11):3197
52. Jiang C, Hu Z, Liu Y et al (2020) A sequential calibration and validation framework for model uncertainty quantification and reduction. Comput Methods Appl Mech Eng 368:113172
53. Kobyzev I, Prince SJ, Brubaker MA (2020) Normalizing flows: an introduction and review of current methods. IEEE Trans Pattern Anal Mach Intell 43(11):3964–3979
54. Dinh L, Sohl-Dickstein J, Bengio S (2016) Density estimation using real nvp. arXiv preprint arXiv:1605.08803
55. Ardizzone L, Lüth C, Kruse J et al (2019) Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392

56. Grover A, Dhar M, Ermon S (2018) Flow-gan: combining maximum likelihood and adversarial learning in generative models. In: Proceedings of the AAAI Conference on Artificial Intelligence
57. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Network 18(5–6):602–610
58. Akesson M, Singh P, Wrede F, et al (2021) Convolutional neural networks as summary statistics for approximate Bayesian computation. IEEE/ACM Transact Comput Biol Bioinform
59. Jaishi B, Ren W-X (2005) Structural finite element model updating using ambient vibration test results. J Struct Eng 131(4):617–628
60. Yang J-H, Lam H-F (2018) An efficient adaptive sequential Monte Carlo method for Bayesian model updating and damage detection. Struct Control Health Monit 25(12):e2260
61. Suita K, Suzuki Y, Takahashi M (2015) Collapse behavior of an 18-story steel moment frame during a shaking table test. Int J High-Rise Build 4(3):171–180
62. Jin S-S, Cho S, Jung H-J et al (2014) A new multi-objective approach to finite element model updating. J Sound Vib 333(11):2323–2338
63. Zhu Z, Au S-K, Li B et al (2021) Bayesian operational modal analysis with multiple setups and multiple (possibly close) modes. Mech Syst Signal Process 150:107261
64. Ni P, Li Q, Han Q et al (2023) Substructure approach for Bayesian probabilistic model updating using response reconstruction technique. Mech Syst Signal Process 183:109624
65. Do NT, Gül M (2019) A time series based damage detection method for obtaining separate mass and stiffness damage features of shear-type structures. Eng Struct 208:109914
66. Song M, Astroza R, Ebrahimian H et al (2020) Adaptive Kalman filters for nonlinear finite element model updating. Mech Syst Signal Process 143:106837
67. Dong X, Liu X, Wright T, et al (2016) Validation of wireless sensing technology densely instrumented on a full-scale concrete frame structure. In: Transforming the future of infrastructure through smarter information: Proceedings of the International Conference on Smart Infrastructure and Construction Construction, 27–29 June 2016. ICE Publishing
68. Dong X, Wang Y (2018) Formulation and optimization algorithm comparison for the FE model updating of large-scale models

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.