Multi-task framework for vibration-based structural damage detection of spatial truss structure using graph learning

Viet-Hung Dang[1*] and Huan X. Nguyen[2]

[1*]Department of Structural Mechanics, Hanoi University of Civil Engineering, Hanoi, Vietnam.
[2]London Digital Twin Research Centre, Faculty of Science and Technology, Middlesex University, London, UK.

*Corresponding author(s). E-mail(s): hungdv@huce.edu.vn;
Contributing authors: H.Nguyen@mdx.ac.uk;

Abstract

Purpose The spatial truss is a special type of structure in civil engineering with striking visualization, used for covering large spaces such as stadiums, commercial centers, and train stations. Due to its multi-component nature and large size, timely monitoring of its structural health is a tedious and challenging mission, for which using vibration signals measured from embedded sensors has shown promising results, reducing significant time and effort compared to manual methods. In order to exploit vibration data more effectively, this study explores a novelty data-driven approach that performs multiple structural damage detection tasks ranging from detecting damage, localizing damage, and quantifying damage severity. **Methods** The main steps of the proposed approach are: converting truss data, including geometrical information and vibration signal, into graph data, leveraging the graph attention network for spatial-temporal feature extraction, and elaborating a compound loss function for multi-task learning. The proposed approach's efficiency and efficacy are quantitatively demonstrated via four case studies with increasing levels of complexity. **Results** The results show that the detection achieves more than 95% accuracy for both a 2D truss structure with 23 elements and a 3D dome truss with 120 bars, while around 90% detection accuracy is obtained for multi-damage scenarios in a 3D spatial double-grid truss with 581 bars. Furthermore, the high detection accuracy is reaffirmed with experimental data of an actual truss structure. **Conclusion** The unique advantage of the proposed model over other counterparts is the ability to encode the geometrical configuration of the truss structure via the adjacency matrix. Therefore, it can be applied to various truss structures in a straightforward way with minor adjustments, given appropriate training data.

Keywords: deep learning, stochastic processes, forecasting, structural engineering, numerical simulation

1 Introduction

The spatial truss is a particular structure class with a high stiffness/weight ratio and visually appealing form, regarded as a *de facto* choice for roof structures of very large spaces, for example, Changi Airport in Singapore, Arena stadium in Germany, the largest industrial factory Boeing Everett in the USA, etc. Truss structure typically consists of a large number of members with similar geometrical and material properties, thus ensuring its structural health in a pristine state is a tedious and challenging task, for which using vibration signal measured from embedded sensors has shown promising results, reducing significant time and efforts compared to manual visual inspection methods. Furthermore, with the development of advanced technologies such as Wireless Sensor Networks, Internet of Things, and Big Data, measurement data can be collected continuously and in the long-term, thus catalyzing the evolution of Structural Health Monitoring (SHM) at scale [1]. However, turning massive measurement data into actionable information is a challenging task for the engineering community not only because of the high dimensionality of data but also due to inevitable uncertainties such as computational errors, signal noise, and environmental effects.

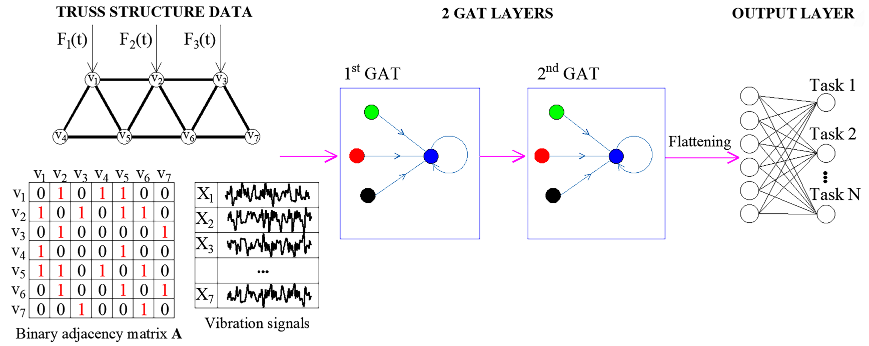
Towards a pragmatism of automatically monitoring the structure's health, dubbed Digital Twin [2], it is required to build an evolving digital replica of the physical structure [3]. The former is closely linked to the latter by a system of sensors measuring the structure's operational state. Conventionally, the digital replica is built using the finite element method with initial input parameters corresponding to the structure's intact state. Then, model updating methods [4] are utilized to adjust these input parameters such that the updated numerical model can reflex the structure's behavior with minimum deviation compared to sensor measurement. After that, the numerical model can assess the structure's current state, detect unobserved damage, and predict future reliability. Although able to provide reliable results and applicable for a wide range of structures, this approach requires significant computational time and is only available to a small portion of users having expertise in structure, optimization, and modeling. Recently, the data-driven method has gained increasing attention from the engineering community thanks to its rapid inference time, highly satisfying results for a spectrum of domains [5, 6].

One of the common ways to build an SHM application is to extract dynamic characteristics such as frequencies, mode shapes, etc., from measured vibration data and then use optimization or machine learning algorithms to detect damage existence. Favarelli and Giorgetti [7] compared the performance of various machine learning-based algorithms using modal frequencies as input for detecting anomalies in bridge structures. Hosseinabadi et al. [8] used wavelet transform as a feature extractor to distill features from vibration time-series data, which were utilized as input for an artificial neural network-based damage detector. Such a method required an user-defined mother wavelet type and associated scale and shift parameters. Santos et al. [9] proposed a novel optimization algorithm called the global expectation-maximization genetic algorithm applicable for non-linear vibration data. Although the feature-based methods can achieve fairly high detection accuracy, they require extracting structures' nature frequencies from raw data in advance via some specialized techniques such as operational modal analysis, stochastic subspace identification method etc. This can hinder its applicability and creditability to a real-time framework, particularly in the case of noisy data, where modal data are sensitive to noise intensity.

Recently, the deep learning-based data-driven technique has gained considerable prominence as it possesses two superior properties: feature learning and scalability. With feature learning, it is able to automatically extract meaningful features from raw data thanks to a hierarchical multi-layer architecture where lower layers identify basic features, and deeper layers synthesize higher-level patterns in terms of learned lower-level ones [10]. With scalability, it is acknowledged that bigger data resort to larger architectures to maximize benefits and vice versa. That is why this artificial intelligent paradigm has become an exciting research direction, achieving impressive results in diverse research areas such as object recognition, machine translation, cancer detection, fault diagnostics, and so forth. In the context of SHM, Avci et al. [11] engineered an approach using 1D convolutional neural network architecture and wireless sensor networks for determining the loss of connection stiffness of a steel frame structure. Later, Zhang et al. [12] also adopted the 1D convolutional neural network method for carrying SHM tasks of a steel bridge structure using vibration sensor signals as input. For monitoring the structural health of plates of different materials, Zhang et al. [13] proposed to use a deep convolutional neural network along with guided waves. Another deep learning architecture was proved efficient in detecting faulty behavior is Long Short Term Memory, which was utilized by Yuan et al. [14] in estimating the remaining useful life of aero-engines under various degradation scenarios. A qualitative and quantitative comparison of different deep learning architectures for SHM tasks was conducted by the authors in [2, 15, 16]. Another benefit of data-driven approaches is that they can be applied to low-cost devices, thus reducing the budget for long-term monitoring, as demonstrated by Moallemi et al. [17].

Though these deep learning architectures can take into account multiple vibration signals as input, they do not have an explicit mechanism to exploit the inherent spatial correlation of sensor locations. This shortcoming might impede the SHM results because, in reality, the number of sensors is significantly smaller than that of the structural members, and measurement signals are only available at a subset of members. On the other hand, retransforming the truss structure into graph data allows for implicitly capturing their geometrical and topological properties. After that, a special architecture in the artificial intelligence literature, namely, the Graph Neural Network (GNN) proposed by Scarselli et al. [18], could be used to exploit this graph information. Though originally, GNN was designed for non-Euclidean data, some authors have proved that it could be beneficial for Euclidean data as discussed in [19], where signal data and connections between signals could be reformulated as a graph, then the fault diagnosis problem can be recast as a graph-level task. Tong et al. [20] reframed the fault detection of transmission lines into a graph classification problem, then employed a GNN model to improve the detection accuracy. Since a truss structure typically has a large number of similar members, it is required to engineer an effective SHM approach able to detect and locate damages with reduced computational time given limited available data from strategically-placed sensors. Therefore, this study investigates a multi-task approach specifically designed for structural damage detection (SDD) of truss structures based on a variant of GNN, namely the graph attention network (GAT). The proposed approach is abbreviated as m-SDDG. Unlike the reviewed works where different detection tasks were carried out separately, one task each time, this approach performs multiple structural damage detection tasks simultaneously. In other words, its output is not a single value but a tuple including a binary value signaling damaged/undamaged states, a value/vector indicating the damaged member's location, and a value/vector in percentage denoting the damage severity. This is done thanks to the integration of three following advents: i) converting truss data including geometrical information and vibration signal into graph data; ii) leveraging the graph attention network for spatio-temporal feature extraction; and iii) elaborating a compound loss function for multi-task learning (MTL). In short, the main contributions of this study can be summarized as follows:

- A data-driven framework is developed for structural damage detection, parallelly performing multiple tasks, including damage detection (SHM level 1), damage localization and damage severity (level 2/3) without significantly increasing computational complexity compared to conventional single-task counterparts (only under 10% more central processing unit (CPU) time).
- The proposed approach is applied to different truss structures under time-varying excitations and shows highly satisfied detection results (>95% accuracy) without compromising any measurement metrics for each task of interest.



	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	1	1	0	0
v_2	1	0	1	0	1	1	0
v_3	0	1	0	0	0	0	1
v_4	1	0	0	0	1	0	0
v_5	1	1	0	1	0	1	0
v_6	0	1	0	0	1	0	1
v_7	0	0	1	0	0	1	0

Fig. 1 Graphical representation of the proposed m-SDDG framework.

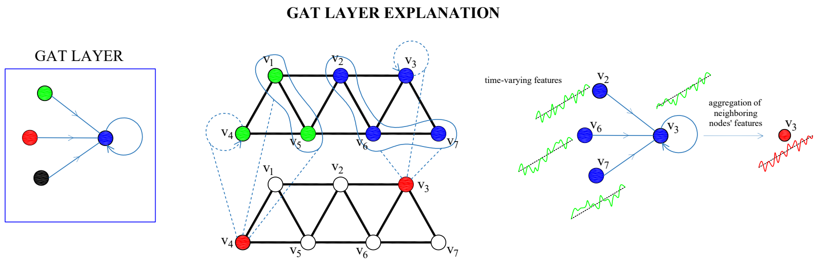


Fig. 2 Explanation of a typical GAT layer.

- Not being limited to the single-damage case, the proposed approach is also able to extend to multi-damage cases while still providing reasonable results (≈90% for damage detection and more than 80% for damage severity).

The rest of the paper is organized as follows: Section II presents the key components of the framework involving the graph attention network-based architecture, and multi-task learning. In Section III, the performance of the proposed approach is justified through four case studies. Finally, the conclusions and perspectives are drawn in Section IV.

2 Multi-task SDD Framework using Graph Attention Network

2.1 Vibration-based Structural Health Monitoring

At first, it is relevant to clarify why vibration signals contain meaningful information for damage detection. A set of sensors such as accelerometers or LVDTs are placed at truss joints to measure vibration responses caused by time-varying excitations; the measured quantities can be displacement $U(t)$ or accelerations $\ddot{U}(t)$. Usually, vibration signals measured when the structure is in a healthy state are given in advance. After that, any new vibration signals or features extracted from signals are compared with those collected in the original healthy state to assess the structure' actual status. However, directly comparing raw high-dimensional vibration signals against each other often does not provide satisfactory results. Therefore, current SDD methods involve extracting eigen frequencies and eigenmode shapes from vibration signals by using the operational modal analysis method [21]. This is because the eigen frequencies and eigenmode shape are directly related to the matrix stiffness of the structures via the matrix eigenvalue problem as follows:

$$[\boldsymbol{K} - \omega_i^2 \boldsymbol{M}]\phi_i = 0, \tag{1}$$

where ϕ_i and ω_i are the eigenmode and eigenfrequency of mode i. Thus, calculating the deviation between those values from intact states and those from current states, enables us to estimate the reduction in stiffness matrix and then perform the damage detection. However, the operational modal analysis method has a high-entry level, as it requires expertise in structural dynamics and mathematics. In addition, it needs specialized software such as MACEC [22], making it difficult to incorporate into an online SHM application. Therefore, it is desirable to design another strategy that can effectively handle vibration signals.

In this study, one presents a data-driven approach for performing multiple structural damage detection tasks for truss structures, namely m-SDDG, as illustrated in Fig. 1. The proposed method directly uses vibration signals and the adjacency matrix encoding spatial correlations of sensors as input to perform structural damage detection, bypassing the need for sophisticated feature analysis and additional preprocessing steps such as modal analysis for extracting dynamic characteristics or short-time Fourier transform for spectrogram images of signals. The graph attention layers act as an automatic feature extractor whose outputs are fed into a fully connected layer for performing SDD tasks. The performance of the proposed approach is evaluated via a measurement metric such as accuracy and is graphically presented via a confusion matrix or evolution curves.

Given a structure instrumented with N_s sensors across its body, data collected from each sensor are stored in the form of a vector $U_i \in R^{L_s}$ with L_s being the signal length. Hence, data collected by all N_s sensors are $[U_1, \ldots, U_{N_s}] \in R^{N_s \times L_s}$. Next, the vibration-based SHM approach aims to assess the current operational state of the structure using only U. A structure's state could be denoted by a binary value (0/1) for (healthy/damaged), or an integer value indicating the location of the damaged element, or a categorical variable for damage severity such as minor/moderate/severe. The input and output data of multi-task vibration-based SHM can be aggregated into a database $D = [\boldsymbol{U}, Y_{det}, Y_{local}, Y_{sev}]$ with $\boldsymbol{U} \in R^{N_{sample} \times N_s \times L_s}$ are a 3D tensor of collected data, N_{sample} is the total number of samples, $Y_{det}, Y_{local}, Y_{sev}$ are the N_{sample}-length vectors indicating structures' states for damage detection, localization, and severity, respectively. On the other hand, the database for the single-task model is obtained by considering only one type of output, i.e., $[\boldsymbol{U}, Y_{det}]$, $[\boldsymbol{U}, Y_{local}]$ and $[\boldsymbol{U}, Y_{sev}]$.

With the m-SDDG method, one requires in advance a training dataset that has a different set of vibration signals with corresponding structural states. A structural state can be labeled as "healthy", or "damaged" for damage detection, "minor", "medium", "severe" for damage severity, or a label denoting the damage location for damage localization. Next, the proposed method will learn latent features that are significantly more sensitive to structural states than raw vibration signals. Given a new vibration signal, the learned model will map its latent features with the most likely structural states.

2.2 Graph Attention Network Layer

A truss structure can be naturally represented as a graph $G = (V, E)$, where V represents the set of nodes (truss joints), N_V is the number of nodes, E denotes the set of edges (truss bars) [18]. The connectivity of the truss joints can be concisely described via an adjacency matrix $\boldsymbol{A}$ with binary values. If there is a connection between node i and j then $A_{ij} = 1$, otherwise $A_{ij} = 0$. One refers $ne(v)$ as a set of neighbor nodes connected to node v (including v) and $co(v)$ as edges connecting v with $ne(v)$. In practice, a truss structure can comprise a large number of truss joints, but each joint usually connects to no more than 10 other joints. Hence, matrix $\boldsymbol{A}$ can be represented as a sparse matrix, helping improve memory efficiency.

On the other hand, each node v of a graph has its own set of features denoted by $X_v \in \mathbb{R}^{L_s}$ with L_s being the feature dimension. In this study, the considered features are vibration signals recorded at the truss nodes.

GAT is a method to learn new representations of nodes which are sensitive to the investigated tasks, based on adjacency matrix $\boldsymbol{A}$, node features and attention mechanism. A new representation of node v at layer $l+1$ can be derived from its representation at previous layer l as follows:

$$h_v^{l+1} = f(h_{ne(v)}^l) = \sigma\left(\sum_{u \in ne(v)} \alpha_{u,v} \boldsymbol{W} h_{ne(v)}^l\right) \tag{2}$$

where h^{l+1} and h^l are hidden states of nodes at layer $l+1$ and l, respectively. f is a neighborhood aggregation function. Note that, the first hidden state $l = 0$ of a node corresponds to the input layer, i.e., $h_v^0 = X_v$. In the right-hand side of the equation, $\boldsymbol{W} h_{ne(v)}^l$ is a linear transformation of node representation via a learnable weight matrix $\boldsymbol{W}$. Using $\boldsymbol{W} h_{ne(v)}^l$ rather $h_{ne(v)}^l$ can effectively reduce the dimension of node features, especially in case of lengthy vibration signals before applying the attention mechanism because the latter is quadratic computation and memory consumption with respect to the input length. $\alpha_{u,v}$ is a normalized attention coefficient between two nodes u and v, signifying how much influence the features of node v have on the representation of node

u. More specifically, $\alpha_{u,v}$ is calculated as follows:

$$\alpha_{u,v} = \frac{\exp(e_{u,v})}{\sum_{w \in ne(v)} \exp(e_{w,v})} \tag{3}$$

where $e_{u,v}$ is an unscaled attention coefficient obtained from a single-layer feedforward neural network using the combination of representations of nodes u and v as input:

$$e_{u,v} = a(\boldsymbol{W}h_u^l, \boldsymbol{W}h_v^l), \tag{4}$$

where $a : \mathbb{R}^{D_V} \times \mathbb{R}^{D_V} \to \mathbb{R}$ is a single-layer feedforward neural network, D_V is the feature dimension of $\boldsymbol{W}h_u^l$. It is worth mentioning that the above calculation procedure involves only neighbor nodes $ne(v)$, not the entire graph, and we can compute attentional coefficients for different nodes in a parallel manner, thus, considerably improving the computation efficiency of GAT. Fig. 2 provides a graphical explanation of the GAT layer via a truss structure with 7 nodes. For example, considering node 3 of the truss, their neighbor nodes are 2, 6 and 7, and their vibration signals are v_3, v_2, v_6 and v_7, respectively. Next, using Eq. (2), the new representation of node 3 will be obtained on the rightmost of the figure. A similar procedure is applied for all nodes of the truss structure.

2.3 Multi-task Structural Damage Detection Framework

The MTL model in this study is developed by using the hard parameter sharing strategy. The model consists of intermediate layers with shared parameters for synthesizing latent representation from input data and a task-specific output layer comprised of multiple neurons, each corresponding to an individual task. Traditional damage detection approaches solve separately different structural damage tasks, i.e., damage detection, damage localization, and damage quantification by using different detection models, although their inputs are mostly the same. On the other hand, recently, machine learning scientists have experimentally shown that combining different tasks into a MTL model [23] can, in fact, improve performance for each individual task of interest while simplifying the learning process as there is only one model to train. The superiority of the multi-task approach is attributed to the fact that instead of only using some domain-specific exploited features for overfitting a specific task, the MTL is forced to learn more generic patterns from data to achieve good overall accuracy for all tasks investigated. Similar to the human learning process, knowledge acquired for one task can benefit others, even if they are seemingly unrelated. Being inspired by this finding, one recasts structural damage detection tasks into equivalent machine learning problems, namely, damage detection into binary classification, damage localization and damage severity into multi-label classification. After that, these problems are solved simultaneously by using a compound loss function involving different terms as follows:

$$\Omega(X) = \lambda_D L_D(X) + \lambda_L L_L(X) + \lambda_S L_S(X) \tag{5}$$

where $\Omega(X)$ is the compound loss, X is input data, L_D, L_L, and L_S are losses for damage detection task, damage localization, and damage severity, respectively. λ_L, λ_D, and λ_S are their corresponding weights. These SDD tasks, considered here, are not entirely independent but hierarchically related together. To be more specific, one can only localize damaged members if the damage occurs and then estimates how much their cross-section losses are. Such a hierarchical relationship is accounted for via the values of weights associated with the loss functions of these tasks. These three weights sum up to 1. More specifically, the weight λ_D for the damage detection task is set to 0.5, while those for localization and severity tasks are both 0.25 unless otherwise stated. It can be seen that if the training error related to the localization task is at the highest, the compound loss will be larger than that of two single-task models for damage and severity detection but smaller than that of the localization detection. In the other extreme case, where the tasks are completely unrelated, a negative transfer problem may occur. This means a task may dominate the training process [24], guiding the model to learn some specific representation of data, making the model achieve high accuracy on this task but severely degrade the overall performance on other tasks.

With such a hard sharing-parameter architecture, the number of trainable parameters of a MTL model is only higher than that of single-task models by a small amount resulting from the task-specific fully connected layers. Thus, theoretically, its model complexity and computational complexity could be slightly higher than the most complex task but significantly lower than the total cost of all tasks. The input data consist of N_v vibration signals with length L_s and a 2D adjacency matrix of size $[N_v, N_v]$. The first GAT layer results in new representations of nodes with the shape of $[N_v, L_1]$ where L_1 is a customizable parameter. The number of trainable parameters of this layer is $(L_2+1)\times L_1+2L_1$, where the first term corresponds to the number of parameters of the linear transformation $\boldsymbol{W}h_u$, and the second term $(2L_1)$ correspond to a feed-forward neural network calculating attention coefficients per Eq. (4). Similarly, the detail of the second GAT layer is also clarified in the table. After that, the new representations of nodes are flattened into a 1D vector, before going through a multi-output fully-connected layer, whose number of parameters is proportional to the number of tasks and number of scenarios. In this study, the typical length of vibration signals L_s is from 1024 to 3000, L_1 and L_2 are set to 512 and 128, respectively.

The proposed m-SDDG model is implemented with the help of the deep learning library Pytorch geometric [25] written in Python. In summary, the MTL investigated in this study is a standard hard-sharing, inductive bias MTL where different tasks still belong to the same domain. By using only one MTL model rather than three models for damage detection, damage localization, and damage severity, one could increase the efficiency and reduce storage requirements, thus, making the deployment easier to manage, maintain and update.

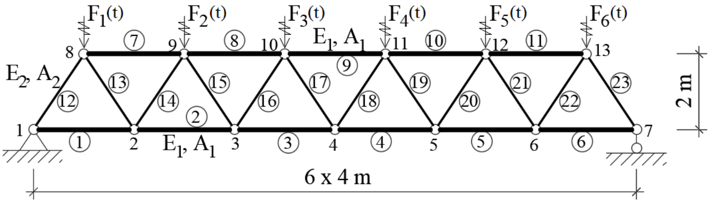


Fig. 3 Schematic representation of the investigated Warren-type planar truss with numbering.

However, there may be a trade-off between efficiency and accuracy as the performance of MTL is sensitive to selected tasks, including but not limited to task difficulty, task complexity, task relatedness, etc.

3 Case Study

In this section, the proposed approach is applied to four case studies with increasing complexity, from a 2D numerical truss with 23 elements to a 3D dome truss with 120 bars, to a 3D spatial truss with 581 bars, and an actual truss structure. Its realization steps are explicitly explained in detail, including structure modeling, data preparation, damage scenarios, training process, and testing detection results.

3.1 Case Study 1: Planar Truss Structure

The first case study aims to validate the applicability of the proposed approach is a Warren-type planar truss, as shown in Fig. 3. The structure consists of 23 bars subjected to time-varying concentrated loads at six nodes on the top chord, replicated from the work of [26]. The total span of the truss is 24 m long, the height is 2 m, Young's modulus is E = 210 GPa, the cross-section area of the top and bottom chords is 20 cm^2 while that of the diagonal bars is 10 cm^2. The external loads here are described by using Gaussian white noise processes, which are widely adopted in the structural literature [27] with amplitudes varying over time. In other words, the amplitude at each time instants is regarded as an independent and identically distributed random variable following a Gaussian distribution. The truss structure is modeled with the help of the finite element software Abaqus Standard [28] with its bars being modeled by using 2-node linear truss element T3D2.

Table 1 Random parameters for Monte Carlo simulations of the planar truss structure.

Parameter	E (MPa)	F (N)	**Damage** **location**	**Damage** **severity** (%)
Distribution	Normal	Normal	Uniform	Uniform
Value	N(210, 21)	N(100, 10)	[1 - 23]	[0, 30, 60, 90]

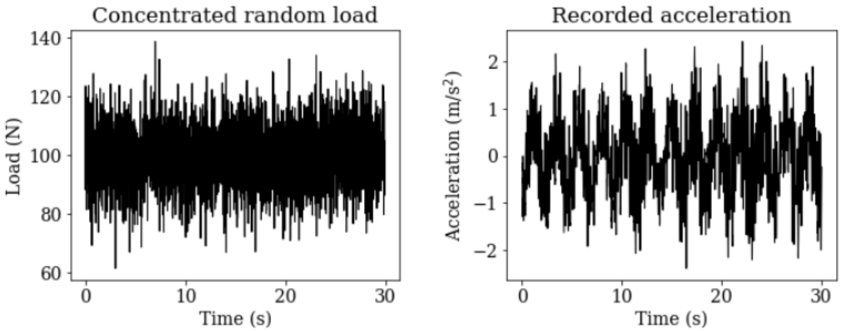


Fig. 4 Representative example of random concentrated load F_3 (left), and corresponding acceleration signal recorded at Node 4 (right).

For this example, one aims to utilize the proposed m-SDDG model to inversely detect damages in the truss structure given vibration signals recorded at truss joints. Fig. 4 illustrates a representative example of a vibration signal obtained at the middle bottom node 4 by using the finite element model. The database for training and validating the m-SDDG method is generated via the Monte Carlo Simulation method. There are 5000 different simulations in total. For each simulation, one randomly introduces damages under the form of section loss, with various severity levels (30%, 60%, and 90% loss), to an arbitrary truss bar. Specifically, there are approximately 200 samples (5000/23) for each damage localization and 1250 cases (=5000/4) for each damage level, as summarized in Table 1. The duration of a simulation is 30 s, and accelerations at every node except two ends are virtually recorded with a sampling frequency $f = 100$ Hz, resulting in 3000-length vibration signals. Hence, the shape of the database is [10000, 11, 3000] where 10000 represents the number of samples, 11 is the number of truss joints (except for two joints with boundary conditions) and 3000 is the signal length. The data are then separately divided into three sub-datasets, namely, training, validation, and testing data in a ratio of 80:10:10. Such a ratio commonly used in the SHM application [29, 30] is also applied to other examples in this study unless otherwise stated.

Next, the training process is carried out, and obtained results are reported. The adopted hyperparameters are the SGD optimizer, a mini-batch size of 128, and an initial learning rate of 10^{-4}. The training process stops when the validation loss does not decrease after 5 consecutive epoches that is known as early stopping technique. The technique helps increase the efficiency of the training process since further training would not improve the model performance. The final results of the training process are presented in Table 2 where Acc_D, Acc_L, and Acc_S are the accuracy results of damage detection, localization, and severity tasks, respectively. More specifically, the accuracy metric is calculated as the ratio of the number of correctly classified samples to the total samples. It can be seen that the multi-task model achieves high accuracy on the validation dataset in all three tasks, i.e., 98.6%, 95.9%, and 97.5% for damage detection, localization, and severity, respectively.

Table 2 Comparison of SDD results for the first case study obtained by the proposed m-SDDG and single-task counterparts.

Dataset	**Number**	**Memory**	**Multi-task model**			
	of data	(Gb)	Acc$_D$ (%)	Acc$_L$ (%)	Acc$_S$ (%)	CPU time (min)
Train	4000	1.85	99.4	98.3	99.0	72.5
Valid	500	0.21	98.6	95.9	97.5	0.44
Dataset	**Single-task model**					
	Acc$_D$ (%)	CPU time (min)	Acc$_L$ (%)	CPU time (min)	Acc$_S$ (%)	CPU time (min)
Train	98.2	62.8	98.8	62.3	94.5	62.4
Valid	96.6	0.41	96.4	0.40	92.6	0.41

Acc$_D$, Acc$_L$, and Acc$_S$ are the accuracy results of damage detection, localization, and severity tasks, respectively.

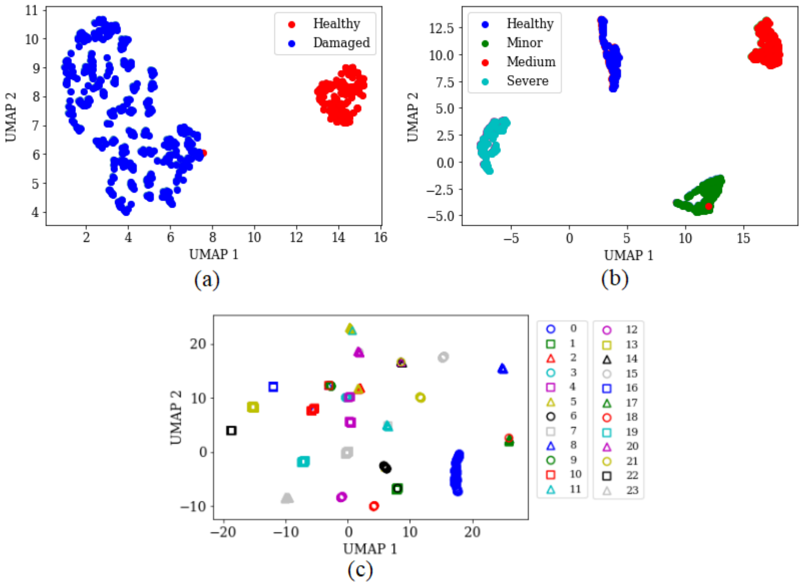


Fig. 5 Graphical representation of detection results via the UMAP technique. a) Damage detection. b) Damage severity. c) Damage localization.

On the other hand, one compares the multi-task model with three single-task counterparts. The single-task model is obtained by adjusting the number of neurons at the output layer to only one and using a single loss term rather than a compound loss involving different terms as described in Eq. (5). Apparently, the table shows that the damage detection and severity accuracies obtained by MTL are clearly higher (98.57% vs. 96.41% and 97.5% vs. 92.6%) except for the localization results (95.9% vs. 96.41%). The reduction in accuracy of some tasks can be explained by the uneven distribution of internal forces within truss bars. For example, minor damage to a truss bar having large internal forces may cause more negative effects on the structure. Meanwhile, bars with lower axial forces that are severely damaged may only slightly reduce the structure's capacity. On the other hand, the training time CPU required by the multi-task model is only around 14% higher than those of single models (73 min vs. 62 min). Furthermore, when performing inference on validation data, the difference in computational time reduces to only about 9% (26.6 s vs. 24.3 s).

To better visualize the detection results on testing data, one utilizes one of the most useful and friendly data visualization techniques, namely, Uniform Manifold Approximation and Projection (UMAP) [31] to graphically represent the similarity among data through clustering and dimension reduction. It is expected that data from the same structural condition will be grouped into the same cluster and well separated from others. Fig. 5 illustrates the UMAP representation of the results on the testing data unseen during the training process with three sub-figures for three detection tasks. Fig. 5(a) shows two well-separated clusters of healthy and damaged class, except for a very small number of red points at the boundary of the blue cluster. Fig. 5(b) also clearly shows 4 clusters for healthy, minor, medium, and severe damage levels, though there still exist some red points (medium) misplaced inside the green cluster (minor). Fig. 5(c) displays all 24 clusters corresponding to 24 possible damage locations. It can be seen that, overall, good separation between clusters is obtained.

In summary, this first case study firmly validates the applicability of the proposed m-SDDG in performing multiple SDD tasks at the same time with accuracy results greater than 95% and time complexity slightly higher than that of single-task competitors (<10%).

3.2 Case Study 2: 3D Dome Truss Structure

Table 3 Random parameters for Monte Carlo simulations of the 3D dome truss structure.

Parameter	E (MPa)	F (kN)	**Damage location**	**Damage severity** (%)
Distribution	Normal	Normal	Uniform	Uniform
Value	N(210, 21)	N(1, 0.1)	[1 - 8]	[0, 30, 60, 90]

For the second case study, the structure of interest consists of 120 steel bars triangularly connected, forming a visually appealing spatial truss with a horizontal span of 3178 cm and a total height of 700 cm, as schematically shown in Fig. 6. The material properties are Young's modulus $E = 210$ GPa, Poisson's coefficient 0.3, volumetric mass density $\rho = 7850$ kg/m^3. The truss bars are partitioned into seven groups, each with seven different cross-section

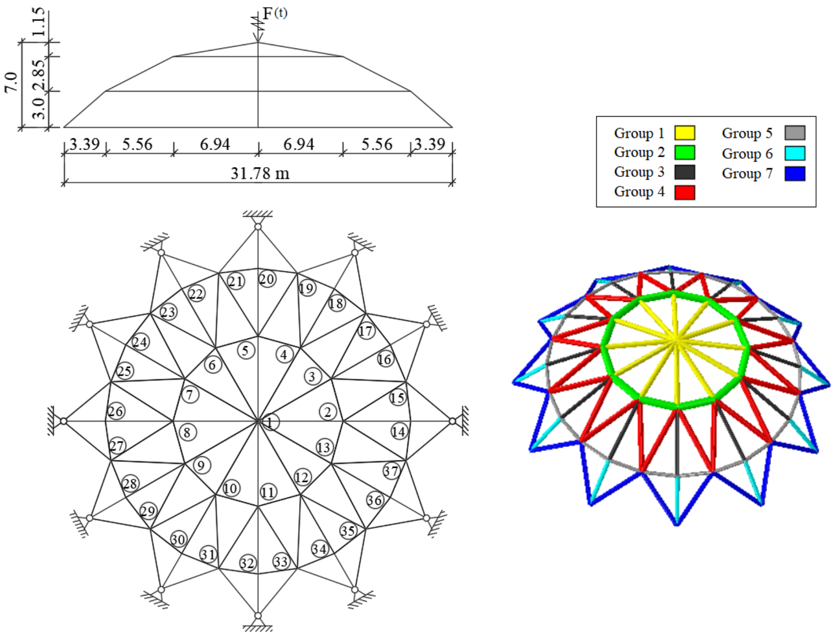
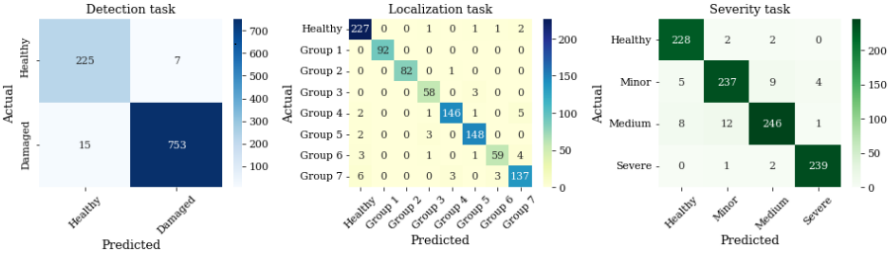


Fig. 6 Schematic representation of the 3D dome truss with (a) dimensions and (b) group labeling.



	Health	Damaged	**Average**
F_1	0.95	0.99	**0.97**

	Health	Group 1	Group 2	Group 3	Group 4
F_1	0.96	1.00	0.99	0.93	0.96

	Group 5	Group 6	Group 7	**Average**
F_1	0.96	0.90	0.92	**0.95**

	Health	Minor	Medium	Severe	**Average**
F_1	0.96	0.93	0.94	0.98	**0.95**

Fig. 7 Confusion matrices of detection results obtained by m-SDDG on the testing dataset.

areas graphically depicted in Fig. 6(a). When performing damage localization, one attempt to identify to which group the damaged element belongs. In terms of excitations, the structure is subjected to a time-varying concentrated load at its summit. The boundary conditions comprise 12 simple supports restraining translation movement in all three directions X, Y, and Z.

For the data generation, a detailed 3D numerical model of the dome structure is first constructed with the help of the software Abaqus. Next, damage scenarios are introduced by reducing the cross-section area of an arbitrary truss bar by an amount in the range of [0, 30, 60, 90]%. There are 10000 simulations in total, and the number of samples for each damage scenario is uniformly distributed as listed in Table 3. Using the same simulation parameters as the first example, i.e., simulation duration $T = 30s$ and sampling frequency $f = 100$ Hz, the shape of the database for the dome truss structure is [10000, 37, 3000] where 37 is the number of truss joints as illustrated in Fig. 6.

To present the proposed approach's performance, one utilizes the confusion matrix, as shown in Fig. 7. The figure shows in detail detection results involving correct results in the matrices' diagonal, including true positive (TP) and true negative (TN) samples and two types of false detections: false-positive one (FP) in the upper off-diagonal cells and false-negative one (FN) in the lower off-diagonal cells. Furthermore, the statistic metric F_1 score is introduced to evaluate the classification accuracy quantitatively, as follows:

$$\begin{aligned} P = & \quad TP/(TP+FP) \\ R = & \quad TP/(TP+FN) \\ F_1 = & \quad 2/(1/P+1/R) \end{aligned} \tag{6}$$

A value $F_1 = 1$ means prediction results in perfect agreement with actual ones. High F_1 scores (≥ 0.95) are achieved by the m-SDDG method, as shown at the bottom of each subfigure. Besides, by adding up diagonal terms of the confusion matrices, the obtained detection accuracy is 978/1000=97.8%. Similarly, the accuracies computed for the localization task and severity task are 95.6% and 95.4%, respectively.

3.3 Case Study 3: Double-Layer Grid Spatial Truss Structure

For case study 3, one considers a double-layer grid truss structure from [32], which is commonly used in covering large public spaces. This structure consists of two parallel planes or curve grids of beams connected together by the third grid of diagonal bars, as shown in Fig. 8. The dimensions of the structure are presented as follows: the horizontal spans in X and Y-direction are $L_x = L_y =$ 18 m, the depth is $h = 3$ m, the number of bars in the top, bottom, and diagonal grids are 100, 81, and 400, respectively, resulting in a total of 581 bars. The steel material has an elastic modulus of 210E3 MPa, yielding stress 235 MPa, Poisson's coefficient 0.3, and volumetric mass density $\rho = 7850kg/m^3$. The cross-sections for the top, bottom and diagonal grid are $D124 \times 4.5$, $D102 \times 4.0$ and $D89 \times 3.5$, respectively. The structure is simply supported at all peripheral joints of the top grid. The time-varying random vertical loads are applied at the inner joints of the top grid. The truss bars are divided into 20 groups as shown in Fig. 8.

In this example, the multi-damage scenario is addressed, which means there is possibly more than one damaged member in the structure. Such scenarios usually happen in reality and pose significant difficulty to conventional methods. In terms of data generation, similar to the previous examples, a Monte

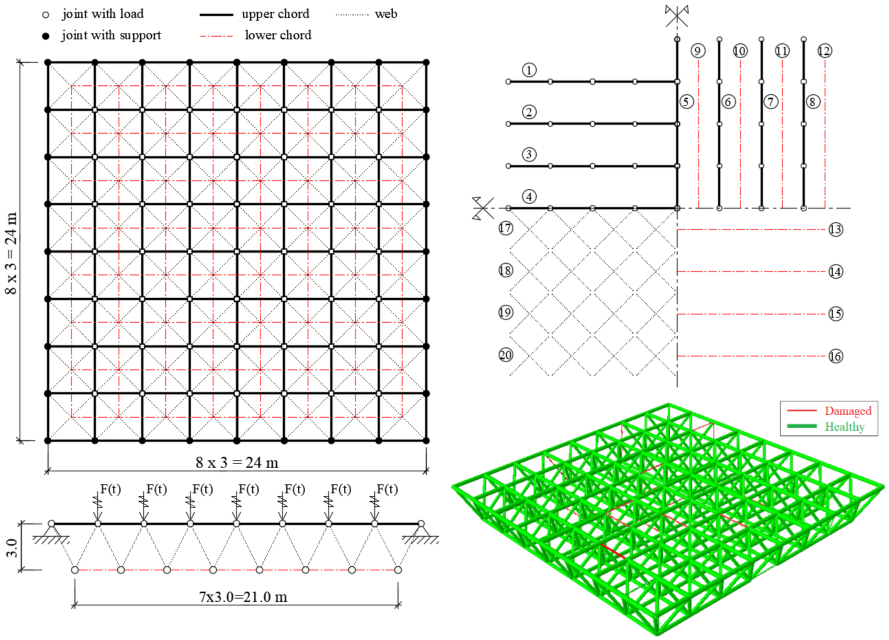


Fig. 8 Schematic representation of the double-layer grid spatial truss structure with dimensions and group labeling.

Table 4 Random parameters for Monte Carlo simulations of the double-layer grid spatial truss structure.

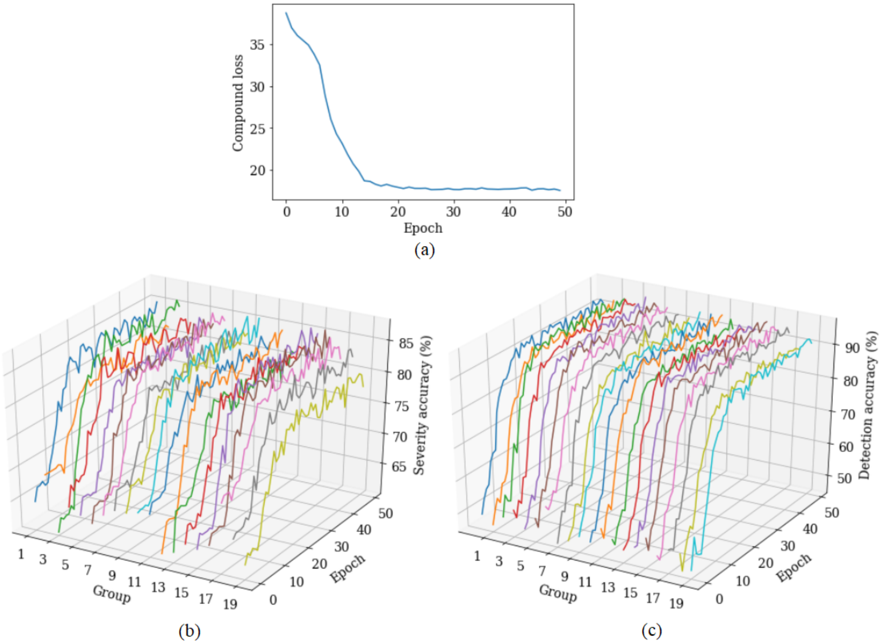
Parameter	E (MPa)	F (kN)	**Damage** **location**	**Damage** **severity** (%)
Distribution Value	Normal N(210, 21)	Normal N(100, 10)	Uniform [1 - 20]	Uniform [0, 30, 60, 90]

Carlo simulation is carried out with 10000 simulations, each having a different number of damaged elements ranging from 1 to 20. For each simulation, accelerations at about 50 joints (one-third of the total joints) at both the top and bottom layers, excluding those with boundary conditions, are recorded and used later as input for the m-SDDG model. Hence, the database has a shape of [10000, 50, 3000].

The proposed multi-task method can handle the multi-damage scenario by modifying the compound loss defined in Eq. (5) as follows:

$$\Omega = \sum_{i=1}^{20} (\lambda_{i,D} L_{i,D} + \lambda_{i,S} L_{i,S}), \tag{7}$$

where the subscript i denotes the structural groups, $L_{i,D}$ and $L_{i,S}$ are losses of damage detection and damage severity for members in group i. $\lambda_{i,D}$ and $\lambda_{i,S}$ are their respective weights. Herein, $\lambda_{i,D} = 0.7/20$ and $\lambda_{i,S} = 0.3/20$, as mentioned in previous examples, the weight for the damage detection task should be more important. Besides, the output of the model is modified to a 40-neuron layer. Each pair of consecutive neurons corresponds to a structural group: one for detecting damage and the other for assessing its severity..



(a)

(b)

(c)

Fig. 9 Learning curves obtained on a) training dataset and b, c) validation dataset.

The SDD results of the proposed m-SDDG model for the 3D spatial truss structure are presented in Figs. 9 and 10, showing how its performance improves throughout the training process as well as what is the final testing accuracy. Fig. 9a illustrates that the training compound loss decreases

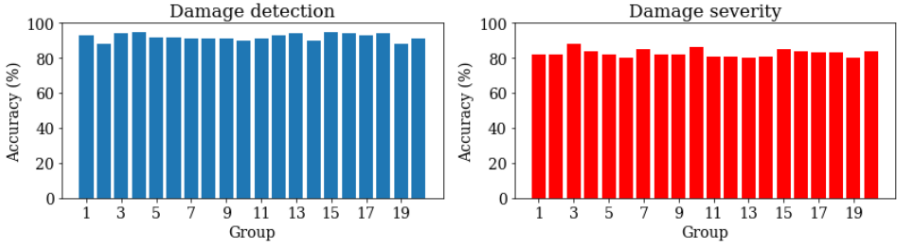


Fig. 10 Accuracy results obtained on testing dataset

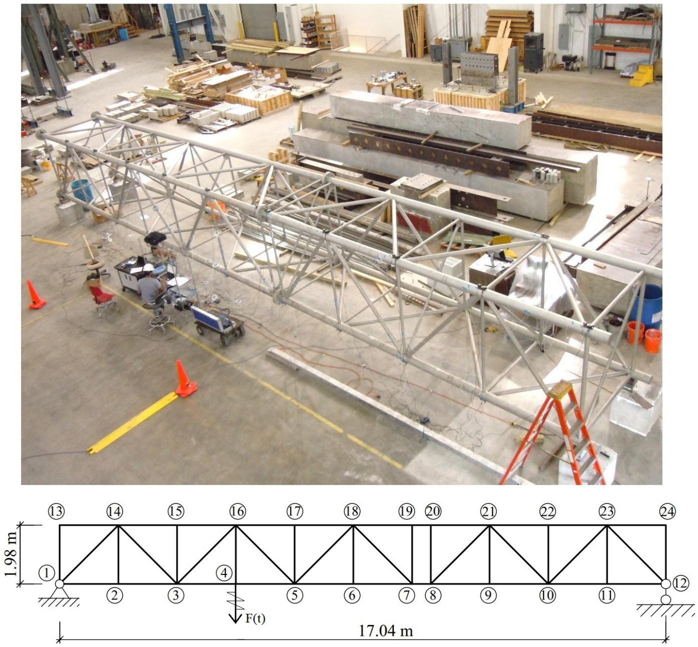


Fig. 11 Aluminum truss structure at Purdue University with node numbering and dimensions [33]

sharply for the first 15 epochs, then diminishes gradually before becoming stable from epoch 30. Meanwhile, the validation accuracies for damage detection and severity for all 20 groups exhibit clear improvements for the period of [0-15] epoch, followed by fluctuating behaviors. The training process finished at epoch near 50 according to the stopping criteria of no improvement in validation loss after 5 consecutive epochs. Final testing accuracy results are plotted through two bar charts in Fig. 10. However, for this example, the accuracy of the severity task is lower than that of the detection task (80% vs. 90%), which can be explained by the highly repetitive nature of the structural layout. Such a nature has a positive effect in reducing the adversity of damaged elements making it more difficult for the m-SDDG model to estimate the extent of damage accurately.

3.4 Case Study 4: Experimental data

For the last case study, the m-SDDG framework was applied to an experimental aluminum truss structure realized at Purdue University [33]. The truss has a length of 17.04 m, a width of 1.83 m, and a height of 1.98 m, consisting of 144 bars of cylindrical cross-sections connected together by 48 bolt joints, as shown in Fig. 11. The structure was excited by band-limited white noise excitation in the vertical direction using an electro-dynamic shaker placed at node 4. Its dynamic responses were measured by using a network of 24 sensors.

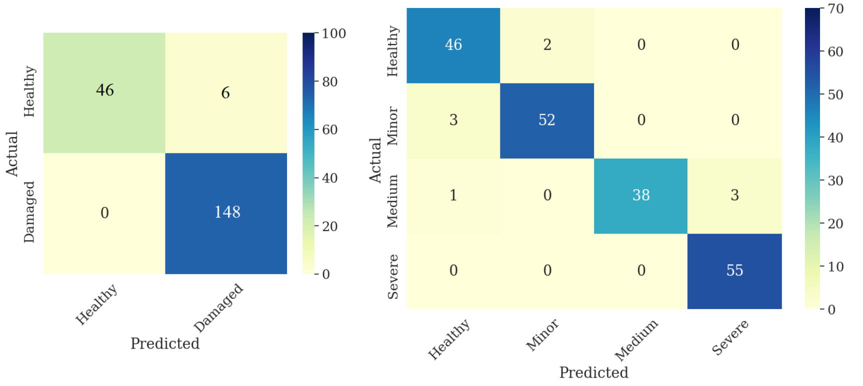


Fig. 12 Confusion matrices of detection results obtained by m-SDDG on the experimental dataset.

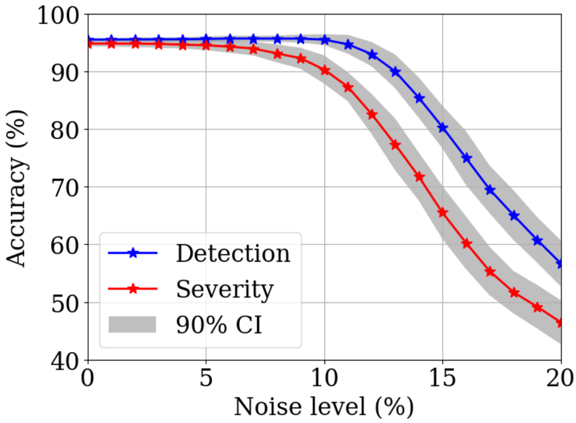


Fig. 13 Evolution of the detection accuracy of the model versus the noise level

The sampling frequency of the sensors is 512 Hz. Damage was introduced into the structure by a cut through one or two truss bars, and the severity of the damage was characterized by the depth of the cuts. The first damage scenario is created by cutting one-third of the cross-section of the bar between nodes 34 and 45, while the second scenario involves cutting up to three-fourths of this bar's cross-section. The third damage scenario consists of this three-fourths cross-section cut and an additional three-fourths cross-section cut of the bar between nodes 6 and 29. These three damage scenarios correspond to minor, medium, and severe states, respectively. The measurement data were then divided into three sub-datasets with a ratio of 60:20:20. Specifically, the shapes of the training, validation, and testing dataset are [600, 24, 1024], [200, 24, 1024], and [200, 24, 1024], where the first number is the number of samples in the datasets, 24 denotes the number of sensors, and 1024 is the length of vibration signals.

Next, the training and validation processes were carried out in a similar way as in the previous examples. For brevity, one solely presents the main results on the testing dataset through Fig. 12. The figure shows the SDD results on experimental data through two confusion matrices. It can be seen that for the damage detection and damage severity tasks, the proposed method achieves results with accuracies of 97.0% and 95.4%, respectively. These results are slightly lower than those obtained on simulation data in examples 1 and 2 (98.6% and 97.8% vs. 97.0% for damage detection). This reduction can be explained by the difference in data volume. Specifically, the total number of simulation samples is 5000, i.e., five times greater than that of experimental data.

In addition, the robustness of the proposed approach against the adverse effect of noise is investigated. One injects white noises with different amplitudes into the original vibration signals and estimates the respective detection accuracy of the m-SDDG framework on noisy data. Mathematically, the noisy data can be expressed as follows:

$$X^{noisy} = X + \alpha \times \epsilon \tag{8}$$

where X and X^{noisy} are original and noisy data, ϵ is a white-noise signal with zero mean and unit variance, α is the noise amplitude measured based on the percentage of the root mean square value of the original signal X. Note that the same trained m-SDDG model presented above is tested with different noise amplitudes. For each noise level, the calculation is repeated ten times, and the mean results and associated confidence interval (mean ± two times standard deviation) are reported. The robustness results are presented in Fig. 13, showing that the m-SDDG can maintain good results for noise levels up to 5%. For noise with a level of more than 10%, the accuracy quickly decreases, and the associated uncertainty of the predicted results also increases as the confidence interval widens correspondingly. The severity task is more affected by the noise than the detection task, e.g., with 20% noise, the accuracy of the former is around 55% which is 10% higher than that of the latter.

4 Conclusion

In this study, a data-driven method for structural damage detection was developed based on Graph Attention Network and multi-task learning. The proposed method is able to address both SHM level 1 task (damage detection), level 2 (damage localization), and level 3 (damage severity) at the same time. Moreover, as the network's parameters are shared between tasks, thus it requires only one running procedure to perform multiple structural damage detection tasks leading to reduced time complexity. Throughout the paper, the general working flow of the proposed approach, as well as the details of its key components involving structure-related graph data, GAT-based architecture, and MTL were discussed; furthermore, realization steps were explicitly demonstrated via various case studies. Besides, a unique advantage of the proposed model over other counterparts is the ability to encode the geometrical configuration of the truss structure via the adjacency matrix. Therefore, it can be applied to various truss structures in a straightforward way with minor adjustments at the output layer, given appropriate training data. Thanks to this flexibility, the proposed method was successfully validated with different structures, as demonstrated via four case studies, proving its practicability. Whereas in the other deep learning and machine learning methods, which do not explicitly take into account the structure topology, even a slight modification such as adding or removing a structural member or change of member orders may considerably affect the model performance. Resultantly, the proposed m-SDDG consistently achieves highly accurate results (>95%) on both numerical and experimental databases of various spatial truss structures. For multiple-damage scenarios, the damage detection tasks still receive satisfying results (>90%) for all structural groups, while for damage severity, reasonable results of ∼80% are obtained. In addition, the m-SDDG is very efficient in terms of time complexity and memory usage as it processes data only one time for different tasks.

As a final remark, simulation and experimental data have their own advantages and inconveniences. Simulation data are generated via ideal models of structures with several assumptions (e.g., frictionless connection, perfectly straight members, negligible geometrical changes, etc.) and without any unexpected factors. In contrast, experiments provide more realistic data, thus allowing for evaluating the model performance more reliably. However, experiments are time-consuming and expensive; thus, the volume of experimental data is limited, and it is impossible to try all damage scenarios on real structures. That is why it is desirable to find an active strategy to train and update the data-drive model with numerical, experimental, and real data of the same structures in an online fashion, paving the way toward a digital twin paradigm.

Appendix

Ablation study

In order to clarify the role of the GAT layer, one carries out an ablation study in which one varies the number of GAT layers from 0 to 3, and observes the corresponding damage detection accuracies on four investigated truss structures. The ablation study results, enumerated in Table 5, demonstrate that using GAT layers significantly improves detection accuracy compared to directly putting data into a dense network classifier without any GAT layer. Moreover, it is found that the model with 2 GAT layers performs better than the model with 1 GAT layer. Meanwhile, a more complex model with 3 GAT layers does not always yield results as good as the model with 2 GAT layers.

Table 5 Ablation study results showing the contribution of the GAT layer on obtained detection accuracy.

Model	**Planar truss**	**Dome truss**	**Double-grid truss**	**Experimental truss**
No GAT layer	77.0%	82.0%	64.6%	83.1%
1 GAT layer	96.1%	93.0%	87.4%	95.1%
2 GAT layer	98.6%	97.8%	90.0%	97.2%
3 GAT layer	91.7%	93.5%	89.9%	97.6%

Learning curves

In order to clarify the progress of the model during the training process, besides the learning curve already presented above for the double-layer spatial truss, the learning curves for the three other examples are presented in Fig. 14. Because the early stopping strategy is used, thus the learning process terminates after five consecutive epochs with no reduction in the compound loss function.

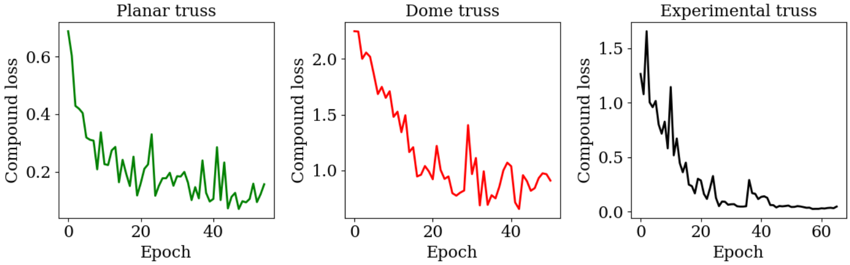


Fig. 14 Learning curves of the m-SDDG model for the planar, dome and epxerimental trusses.

Effect of the number of sensors

When working with large structures such as the double-layer grid spatial truss structure in example 2, installing sensors to all truss joints is impossible. Therefore it is necessary to estimate the impact of the number of sensors on the model's SDD performance. One varies the number of sensors in the range of [1, 10, 15, 30, 50, 90, 120, 145] and carries out a similar procedure training and validation as previously described above, then report the computed results, including detection accuracy, severity accuracy, and CPU time in Table 6. The CPU time investigated in this study is the program's execution time but does not include delay time for waiting for computational resources. It can be seen that using only one sensor cannot provide satisfying results but is still higher than random solutions (57.3% vs. 50.0%). Furthermore, increasing the number of sensors from 1 to 50 improves SDD results significantly, as the detection accuracy clearly increases from 57.3% to 89.7% and severity accuracy from

48% to 80.3%. However, increasing the number of sensors from 50 to 120 only shows a marginal improvement in the obtained results, i.e., by 1.7% from 89.7% to 90.4%. Especially, using vibration signals from all truss joints (145 joints) leads to an over GPU memory error, which is 11 Gb for GeForce 2080i, then no SDD result is available (N/A). On the other hand, the CPU time required for the training process is approximately proportional to the used number of sensors, from 1.8 minutes for one sensor to 54.3 minutes for 50 sensors and 138.1 minutes for 120 sensors. Based on these results, setting the number of sensors to around 50 provides the best performance-efficiency gain-loss estimation for the double-layer grid truss structure under investigation.

Table 6 SDD results versus the number of virtual sensors as input

Number of sensor	training CPU time (m)	Detection accuracy (%)	Severity accuracy (%)
1	1.8	57.3	48.0
10	11.5	72.0	60.1
15	18.0	78.1	71.3
30	27.5	86.2	78.4
50	54.3	89.7	80.3
90	128.0	90.1	80.7
120	138.1	90.4	81.1
145	N/A	N/A	N/A

Data Availability. Source code and processed data are available from the corresponding author upon reasonable request

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

References

[1] Sun, L., Shang, Z., Xia, Y., Bhowmick, S., Nagarajaiah, S.: Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection. Journal of Structural Engineering **146**(5), 04020073 (2020)

[2] Dang, H.V., Tatipamula, M., Nguyen, H.X.: Cloud-based digital twinning for structural health monitoring using deep learning. IEEE Transactions on Industrial Informatics **18**(6), 3820–3830 (2022). https://doi.org/10.1109/TII.2021.3115119

[3] Nguyen, H.X., Trestian, R., To, D., Tatipamula, M.: Digital twin for 5g and beyond. IEEE Communications Magazine (2020)

[4] Friswell, M.I., Mottershead, J.E.: Finite Element Model Updating in Structural Dynamics. Springer, ??? (2001)

[5] He, K., Su, Z., Tian, X., Yu, H., Luo, M.: Rul prediction of wind turbine gearbox bearings based on self-calibration temporal convolutional network. IEEE Transactions on Instrumentation and Measurement **71**, 1–12 (2022). https://doi.org/10.1109/TIM.2022.3143881

[6] Zhou, P., Zhou, G., Wang, H., Wang, D., He, Z.: Automatic detection of industrial wire rope surface damage using deep learning-based visual perception technology. IEEE Transactions on Instrumentation and Measurement **70**, 1–11 (2021). https://doi.org/10.1109/TIM.2020.3011762

[7] Favarelli, E., Giorgetti, A.: Machine learning for automatic processing of modal analysis in damage detection of bridges. IEEE Transactions on Instrumentation and Measurement **70**, 1–13 (2021). https://doi.org/10.1109/TIM.2020.3038288

[8] Hosseinabadi, H.Z., Nazari, B., Amirfattahi, R., Mirdamadi, H.R., Sadri, A.R.: Wavelet network approach for structural damage identification using guided ultrasonic waves. IEEE Transactions on Instrumentation and Measurement **63**(7), 1680–1692 (2014). https://doi.org/10.1109/TIM.2014.2299528

[9] Santos, A., Santos, R., Silva, M., Figueiredo, E., Sales, C., Costa, J.C.W.A.: A global expectation–maximization approach based on memetic algorithm for vibration-based structural damage detection. IEEE Transactions on Instrumentation and Measurement **66**(4), 661–670 (2017). https://doi.org/10.1109/TIM.2017.2663478

[10] Brownlee, J.: What is Deep Learning? (2019). https://machinelearningmastery.com/what-is-deep-learning

[11] Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Inman, D.J.: Wireless and real-time structural damage detection: A novel decentralized method for wireless sensor networks. Journal of Sound Vibration **424**, 158–172 (2018). https://doi.org/10.1016/j.jsv.2018.03.008

[12] Zhang, Y., Miyamori, Y., Mikami, S., Saito, T.: Vibration-based structural state identification by a 1-dimensional convolutional neural network. Computer-Aided Civil Infrastructure Engineering **34**(9), 822–839 (2019). https://doi.org/10.1111/mice.12447

[13] Zhang, B., Hong, X., Liu, Y.: Deep convolutional neural network probability imaging for plate structural health monitoring using guided waves. IEEE Transactions on Instrumentation and Measurement **70**, 1–10 (2021). https://doi.org/10.1109/TIM.2021.3091204

[14] Yuan, M., Wu, Y., Lin, L.: Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In: IEEE International Conference on Aircraft Utility Systems (2016)

[15] Dang, H.V., Raza, M., Nguyen, T.V., Bui-Tien, T., Nguyen, H.X.: Deep learning-based detection of structural damage using time-series data. Structure and Infrastructure Engineering, 1–20 (2020)

[16] Dang, H.V., Tran-Ngoc, H., Nguyen, T.V., Bui-Tien, T., De Roeck, G., Nguyen, H.X.: Data-driven structural health monitoring using feature fusion and hybrid deep learning. IEEE Transactions on Automation Science and Engineering **18**(4), 2087–2103 (2021). https://doi.org/10.1109/TASE.2020.3034401

[17] Moallemi, A., Burrello, A., Brunelli, D., Benini, L.: Model-based vs. data-driven approaches for anomaly detection in structural health monitoring: a case study. In: 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6 (2021). https://doi.org/10.1109/I2MTC50364.2021.9459999

[18] Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE transactions on neural networks **20**(1), 61–80 (2008)

[19] Chen, D., Liu, R., Hu, Q., Ding, S.X.: Interaction-aware graph neural networks for fault diagnosis of complex industrial processes. IEEE Transactions on Neural Networks and Learning Systems, 1–14 (2021). https://doi.org/10.1109/TNNLS.2021.3132376

[20] Tong, H., Qiu, R.C., Zhang, D., Yang, H., Ding, Q., Shi, X.: Detection and classification of transmission line transient faults based on graph convolutional neural network. CSEE Journal of Power and Energy Systems **7**(3), 456–471 (2021)

[21] Reynders, E.: System identification methods for (operational) modal analysis: review and comparison. Archives of Computational Methods in Engineering **19**, 51–124 (2012)

[22] De Roeck, G., Peeters, B., Ren, W.-X.: Benchmark study on system identification through ambient vibration measurements. In: Proceedings of IMAC-XVIII, the 18th International Modal Analysis Conference, San Antonio, Texas, vol. 1106, p. 1112 (2000)

[23] Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 109–117 (2004)

[24] Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7482–7491 (2018)

[25] Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)

[26] Liu, Z., Choe, Y.: Data-driven sensitivity indices for models with dependent inputs using polynomial chaos expansions. Structural Safety **88**, 101984

[27] Li, Z., Lam, C., Yao, J., Yao, Q.: On testing for high-dimensional white noise. The Annals of Statistics **47**(6), 3382–3412 (2019)

[28] Dassault, S.: ABAQUS Analysis User's Manual, (2016)

[29] Abdoli, S., Cardinal, P., Koerich, A.L.: End-to-end environmental sound classification using a 1d convolutional neural network. Expert Systems with Applications **136**, 252–263 (2019)

[30] Xu, Y.-L., Lin, J.-F., Zhan, S., Wang, F.-Y.: Multistage damage detection of a transmission tower: Numerical investigation and experimental validation. Structural Control and Health Monitoring **26**(8), 2366 (2019)

[31] McInnes, L., Healy, J., Saul, N., Grossberger, L.: Umap: Uniform manifold approximation and projection. The Journal of Open Source Software **3**(29), 861 (2018)

[32] Li, H., Taniguchi, Y.: Load-carrying capacity of semi-rigid double-layer grid structures with initial crookedness of member. Engineering Structures **184**, 421–433 (2019)

[33] Sun, Z., Krishnan, S., Dyke, S.: Dynamic Testing of a Full Scale Highway Truss: Vertical Electrodynamic White Noise Shaker Excitation. Network for Earthquake Engineering Simulation (NEES) (2013). https://doi.org/10.4231/D33B5W79N. https://www.designsafe-ci.org/data/browser/public/nees.public/NEES-2011-1013.groups/Experiment-1