DOI: 10.1111/mice.12944

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Elastic structural analysis based on graph neural network without labeled data

Ling-Han Song[1] | **Chen Wang**[1] | **Jian-Sheng Fan**[1,2] | **Hong-Ming Lu**[1]

[1]Department of Civil Engineering, Tsinghua University, Beijing, China

[2]Key Laboratory of Civil Engineering Safety and Durability of the Ministry of Education, Tsinghua University, Beijing, China

Correspondence
Chen Wang, Department of Civil Engineering, Tsinghua University, 100084 Beijing, China.
Email: qtwjy309@163.com

Funding information
The National Natural Science Foundation of China, Grant/Award Number: 52121005; China National Postdoctoral Program for Innovative Talents, Grant/Award Number: BX20220177; China Postdoctoral Science Foundation, Grant/Award Number: 2022M711864

Abstract

Artificial intelligence is gaining increasing popularity in structural analysis. However, at the structural system level, the appropriateness of data representation, the paucity of data, and the physical interpretability of results are rarely studied and remain profound challenges. To fill such gaps, a physics-informed model named StructGNN-E (i.e., structural analysis based on graph neural network [GNN]–elastic) based on the GNN architecture, which is capable of implementing the elastic analysis of structural systems without labeled data, is proposed in this study. The systems with structural topologies and member configurations are organized as graph data and later processed by a modified graph isomorphism network. Moreover, to avoid dependence on big data, a novel physics-informed paradigm is proposed to incorporate mechanics into deep learning (DL), ensuring the theoretical correctness of the results. Numerical experiments and ablation studies demonstrate the unique effectiveness of StructGNN-E against common DL models, with an average accuracy of 99% and excellent computational efficiency. Due to its differentiability, StructGNN-E is promising for bidirectionally linking structural parameters and analysis results, paving the way for a new end-to-end structural optimization method in the future.

1 | INTRODUCTION

The emergence of new-generation artificial intelligence technologies represented by machine learning (ML) and deep learning (DL; Dong et al., 2021; Goodfellow et al., 2016; LeCun et al., 2015) is attracting an increasing number of scholars. These technologies have been applied in civil engineering, particularly in the fields of computational structural analysis and computational material analysis (Amezquita-Sancheza et al., 2020; Sun et al., 2021). It is anticipated that the performance of these technologies will surpass the performance of traditional numerical methods to realize the efficient simulation of engineering structures in the digital world (Lee et al., 2018; Salehi & Burgueño, 2018).

Currently, research on ML/DL-based computation in civil engineering has addressed all levels of analysis scenarios. At the construction material level, Rafiei et al. (2017) presented a deep restricted Boltzmann machine to estimate concrete properties based on mixture proportions. Nguyen et al. (2018) adopted artificial neural networks (ANNs) for predicting the strength of foamed concrete. C. Wang et al. (2022) first introduced the Seq2Seq framework and then the attention mechanism (Wang et al., 2022) to simulate the cyclic behavior of low-yield-point steel. Kang et al. (2021) predicted the strength of steel

fiber-reinforced concrete using 11 ML algorithms and analyzed the influential factors. At the structural member level, Olalusi and Awoyera (2021) focused on the shear capacity of slender RC structures with steel fibers and used Gaussian process regression for prediction. Wakjira et al. (2022) employed ensemble ML to estimate the shear capacity of fiber-reinforced-polymer (FRP) reinforced concrete. Naderpour et al. (2021) utilized a decision tree and ANNs to determine the failure mode of RC columns and achieved exceptional performance. At the structural system level, Rafiei and Adeli (2017) combined ML classification algorithms with an optimization process for earthquake prediction. Zhu et al. (2021) attempted to use ANNs and support vector regression to calculate the buckling load of imperfect reticulated shells, avoiding costly computation in nonlinear analyses. Recently, the deep long short-term memory (LSTM) proposed by Hochreiter and Schmidhuber (1997) model shows great potential in seismic analysis. Zhang et al. (2019) developed an LSTM model to compute the seismic responses of a specific building, and the results agreed with the reference data well. Torky and Ohno (2021) combined a convolutional neural network (CNN) with LSTM to predict the seismic response of an industrial-level building.

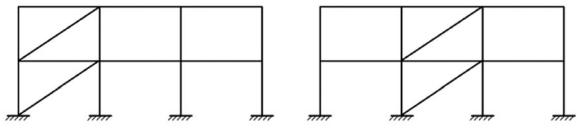
Valuable results have been acquired through relevant studies and have helped engineers efficiently build complicated quantitative relationships in structural computation (Feng et al., 2020; Guan et al., 2021; Graf et al., 2012; Kabir et al., 2021; Oh et al., 2020) and optimization (Hung & Jan, 2002; Messner et al., 1994; Parvin & Serpen, 1999; Yin & Zhu, 2019). However, at the structural system level, existing studies have the following limitations. (1) In the current literature, the data representation of structural systems has rarely been a point of focus, which can possess various topologies due to different structural member configurations and their connectivity, resulting in exponentially increasing complexity, compared to that of materials and structural members. The linear data structure used in current studies cannot comprehensively describe the feature information of structural systems. (2) Applications at this level face severe data scarcity. Most ML/DL models adopt a data-driven paradigm, which relies heavily on big data for learning underlying patterns. Nevertheless, there is little experimental data concerning structural systems, and data generation using classic finite element (FE) analysis is inordinately time-consuming and cannot cover the entire parameter space. It is well-known that common data-driven models suffer from data sparsity (Martins et al., 2020) and are not directly applicable at this level. (3) The theoretical correctness of the analysis results is difficult to guarantee. The inference process of data-driven models often neglects distinctive mechanical backgrounds of structural analysis. Accordingly, researchers and engineers cannot interpret the analysis results (Naser, 2021) and evaluate the correctness of the models based on the test set; this is unacceptable for engineering applications where safety is the primary goal.

To address these limitations, an innovative physics-informed DL model named *StructGNN-E* (i.e., structural analysis based on graph neural network [GNN]–elastic), which is able to implement elastic analysis of structural systems without labeled data, is proposed in this study. The remainder of this paper is organized as follows. In Section 2, we analyze the data representations of structural systems and elaborate on how to utilize non-Euclidean data—the graph—to organize their feature information. On this basis, we develop a GNN architecture to handle the structural analysis of elastic structural systems in Section 3, wherein the basic knowledge of GNNs is introduced and a variant of the graph isomorphism network (GIN) adapting to the structural analysis scenario is illustrated. In addition, we propose a physics-informed paradigm, which integrates fundamental mechanical formulations into DL, to resolve the paucity of data at the structural level. In Section 4, we validate the StructGNN-E model with randomly generated frame structures of different scales. In Section 5, we carry out ablation studies, which further demonstrate the unique effectiveness of the StructGNN-E model, and conceptually discuss the potential application in the field of structural optimization. Finally, we conclude the current work and discuss the outlook of our future work in Section 6.

2 | DATA REPRESENTATION USING GRAPH STRUCTURES

The mechanism of structural analysis can be abstracted as predicting structural responses $\mathcal{R} = (\mathbf{Y}_{t0}, \ldots, \mathbf{Y}_{tn})$ that conform to physical laws given structural properties $\mathcal{S} = (\mathbf{C}_1, \mathbf{C}_2, \ldots)$ and external stimuli $\mathcal{I} = (\mathbf{X}_{t0}, \ldots, \mathbf{X}_{tn})$ (C. Wang et al., 2020). At the structural system level, such as a steel frame structure, the structural properties $\boldsymbol{C}_i$ correspond to the overall spatial topologies, section geometries of underlying steel members, material properties of structural steel, and so forth. The stimulus-response pair $< \boldsymbol{X}_{tj}, \boldsymbol{Y}_{tj} >$ represents structural analysis results such as the load–displacement relationship. The subscript *tj* indicates the stimulus/response at location *j* at time *t*.

To address the problem, traditional methods, including numerical approaches operating at the level of structural matrices and mechanical approaches operating at the level of nodes or FE elements, follow the idea of solving a large problem by decomposing it into a number of small sub-problems and have been proven useful in civil engineering. Methods belonging to the former category



(a) Frame structure No. 1 **(b)** Frame structure No. 2

FIGURE 1 Two examples of frame structures. (a) Frame structure No. 1 and (b) Frame structure No. 2

include singular value decomposition (Pellegrino, 1993), the subspace method (Pellegrino & Calladine, 1986), and group representation theory for symmetric structures (Kangwai & Guest, 2000), which were comprehensively studied before the widespread application of finite element method (FEM) and still play important roles in many algorithms of FE platforms (Lu, 2009). The latter approaches are now the most common approaches for mechanical analysis, ranging from simulations of material behaviors to computations of large-scale structure systems (Reddy, 2019; Zienkiewicz & Robert, 2005).

However, high requirements on mathematical and mechanical knowledge of matrix manipulation, time consumption in building a reasonable FE model, and barriers between different computation platforms make these approaches sometimes unsatisfactory for novel applications. From another perspective, our goal of structural analysis is to build a mapping from the input conditions $(\mathcal{S}, \mathcal{I})$ to the output responses $\mathcal{R}$ based on artificial intelligence techniques.

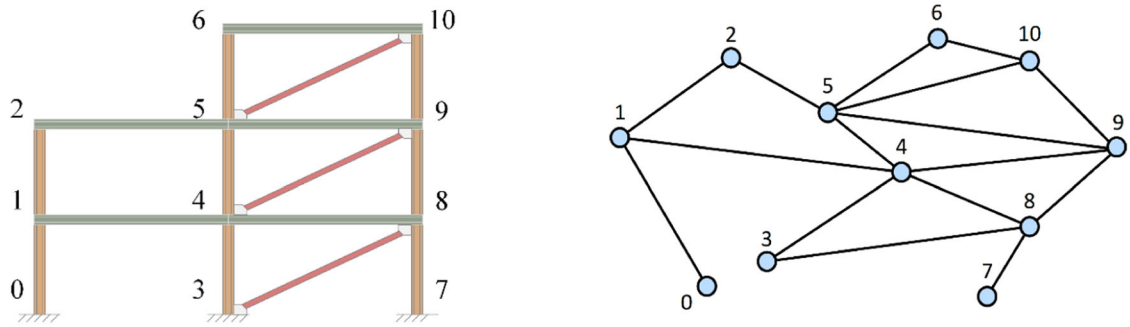
Very few intrinsic structural properties have been described in existing studies (Esteghamati & Flint, 2021; Morfidis & Kostinakis, 2017; Hwang et al., 2021), and in some studies, only different loading protocols were considered (Huang & Chen, 2021; Kim et al., 2019; Lagaros & Papadrakakis, 2012), resulting in a great loss of topological information and thus poor generalization capability across different structures. Reviewing classical numerical studies and engineering design experience, the feature information of structural systems involves two aspects. (1) Structural member connectivity: A structural system comprises numerous structural members, whose location information and connectivity constitute the most intuitive characteristics of the system. Figure 1 shows two frame structures with the same geometric outlines and structural members. However, due to different topological connectivity in the first two spans, these two frame structures exhibit disparate internal force distributions under the same loading conditions, demonstrating the importance of connectivity in structural analysis. (2) Structural member configurations: The configurations of underlying structural members, including what member types to use and their intrinsic properties, significantly affect the overall mechanical responses of the structural system. For example, larger member sections lead to higher structural stiffness. Structural members also serve as the physical carriers of external stimuli $\mathcal{S}$.

Accordingly, compared to Euclidean data structures (such as vectors, sequences, and grids), the feature information of structural systems exhibits two properties. (1) Nonsequential: We cannot define a proper sequential relationship among the structural members, and a physical start or end does not exist. Therefore, DL algorithms that address sequences such as the recurrent neural network (RNN) family are not applicable at the structural system level. (2) Translation variant: Due to architectural functionalities and diverse member placements, structural systems usually have topologically different structures at different locations; thus, organizing the feature information into grids or vectors resembling images or text is not feasible. Moreover, the scale of structural systems varies greatly, and it is difficult to define a metric of Euclidean distance between two samples. In summary, common data structures used in existing studies are incapable of comprehensively describing structural systems; thus, a more advanced data representation method is necessary.

Here, we introduce the graph structure, a type of non-Euclidean data (Zafeiriou et al., 2022), to represent structural systems and capture their distinctive properties. Graphs are a kind of data structure consisting of nodes and edges that characterize objects and their relationships, respectively. Recently, graph structures have received increasing attention in various fields, such as social networks in recommender systems (Ying et al., 2018), molecular modeling in chemistry (Wang et al., 2022), and protein interactions in biology (Tsubaki et al., 2019), due to their extraordinary expressive power. These interdisciplinary explorations have inspired us to propose an innovative data paradigm for structural systems. In general, graph structures are classified into directed graphs and undirected graphs in terms of edge orientations. Considering that it is difficult to predetermine the force transfer paths inside a structural system, we adopt undirected graphs:

$$G = (V, E) \tag{1}$$

where G represents a structural system sample, V are nodes (vertices) representing structural joints, and E are edges that correspond to structural members connecting to the joints. Figure 2 shows the graph counterpart of a planar frame structure with braces (the numbers marked in the figure indicate node identity (i.e., node ID) rather than the sequential relationship of nodes), wherein the nodes are beam–column joints, and the edges are beams, columns, and braces. Let $v_i \in V$ denote a node and $e_{ij}=(v_i, v_j) \in E$



(a) Schematic illustration of a planar frame structure **(b)** Graph representation

FIGURE 2 Graph representation of a planar frame structure. (a) Schematic illustration of a planar frame structure and (b) graph representation

denote an edge connecting v_i and v_j. The neighborhood of a node v is defined as

$$N(v) = \{u \in V \,|\, (v,u) \in E\} \tag{2}$$

One of the most commonly used graph representations is an adjacency matrix $\mathbf{A}$ with $\mathbf{A}_{ij} = 1$ if $e_{ij} \in E$ and $\mathbf{A}_{ij} = 0$ if $e_{ij} \notin E$. To save storage space, we can extend the adjacency matrix $\mathbf{A}$ to record structural properties such as section geometries and stiffness by utilizing the symmetry of undirected graphs. For example, the extended adjacency matrix of the planar structure in Figure 2 can be written as

$$A = \begin{bmatrix}
-1 & EI_{0,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
EA_{0,1} & 0 & EI_{1,2} & 0 & EI_{1,4} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & EA_{1,2} & 0 & 0 & 0 & EI_{2,5} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & EI_{3,4} & 0 & 0 & 0 & EI_{3,8} & 0 & 0 \\
0 & EA_{1,4} & 0 & EA_{3,4} & 0 & EI_{4,5} & 0 & 0 & EI_{4,8} & EI_{4,9} & 0 \\
0 & 0 & EA_{2,5} & 0 & EA_{4,5} & 0 & EI_{5,6} & 0 & 0 & EI_{5,9} & EI_{5,10} \\
0 & 0 & 0 & 0 & 0 & EA_{5,6} & 0 & 0 & 0 & 0 & EI_{6,10} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & EI_{7,8} & 0 & 0 \\
0 & 0 & 0 & EA_{3,8} & EA_{4,8} & 0 & 0 & EA_{7,8} & 0 & EI_{8,9} & 0 \\
0 & 0 & 0 & 0 & EA_{4,9} & EA_{5,9} & 0 & 0 & EA_{8,9} & 0 & EI_{9,10} \\
0 & 0 & 0 & 0 & 0 & EA_{5,10} & EA_{6,10} & 0 & 0 & EA_{9,10} & 0
\end{bmatrix} \tag{3}$$

where each diagonal element $\mathbf{A}_{ii}$ indicates the boundary condition of node v_i (-1 for a confined node); each element in the upper triangle $\mathbf{A}_{ij,i<j}$ represents the bending rigidity of the structural member between node v_i and node v_j; and each element in the lower triangle $\mathbf{A}_{ij,i>j}$ represents the axial rigidity.

Apart from the fundamental representation above, both the nodes and edges can have attributes $\mathbf{X}^v \in \mathbb{R}^{n \times d}$ (n is the number of nodes and d is the dimension of node features) and $\mathbf{X}^e \in \mathbb{R}^{m \times c}$ (m is the number of edges and c is the dimension of edge features) to enhance the expressiveness of the graph. For structural systems, the coordinates of joints are important node features that determine the physical positions of each structural member and their spatial information, such as their lengths. As the subsequent sections will show, node features of coordinates play a critical role in structural analysis.

Based on the concepts and notations of graph data, we can properly digitalize arbitrary structural systems and preserve their feature information with high fidelity, facilitating intelligent structural analysis by downstream DL models.

3 | STRUCTGNN-E MODEL

In practice, most engineering structures adopt elastic analysis solutions under static loads during the design stage. Furthermore, elastic cases also serve as the foundations of elasto-plastic cases, which provide key techniques and help develop the general framework. Therefore, this study takes elastic structural systems under static loads as the research object.

Adapting to the graph data representation, we propose a DL structural analysis model named *StructGNN-E* (i.e., structural analysis based on GNN–elastic) within the GNN framework. The model architecture is shown in Figure 3. The feature information of structural systems is transformed into graph data and sent to a variant of GIN, which computes the internal force distribution using a message-passing mechanism. The whole model is driven by structural mechanics without labeled data, overcoming the data scarcity problem at the structural system level.

3.1 | Basics of GNNs

GNNs are a kind of emerging DL technique that extends classical neural networks such as CNNs and RNNs to process non-Euclidean graph data. A fundamental assumption underlying GNNs is that the targets for prediction

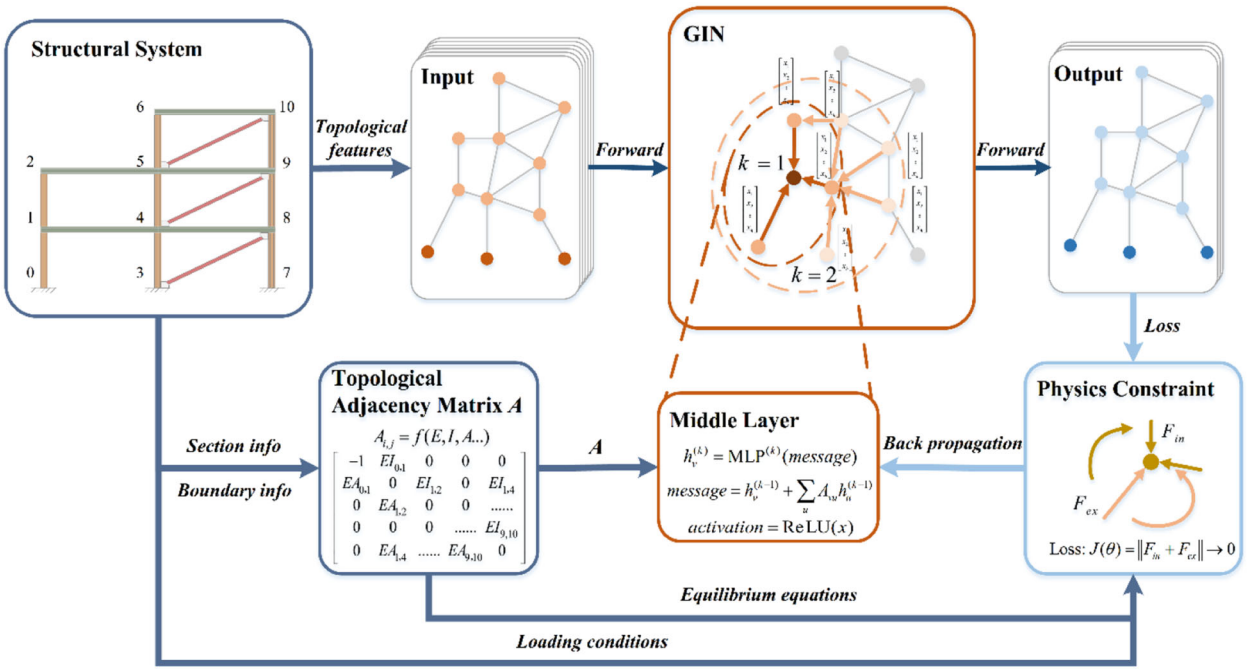


FIGURE 3 Architecture of the StructGNN-E (structural analysis based on graph neural network [GNN]–elastic) model. GIN, graph isomorphism network

should be invariant to the order of graph nodes. Therefore, GNN parameters are independent of the node ordering and are shared across the entire graph. In general, GNNs use the graph structure along with node and edge features for representation learning of nodes or the entire graph. The framework of most modern GNNs can be formulated using a message-passing mechanism (Zhou et al., 2020), wherein an L-layered GNN is expressed as:

$$\boldsymbol{h}_v^{(0)} = \boldsymbol{x}_v, \forall v \in V \tag{4}$$

$$\boldsymbol{m}_{uv}^{(l)} = \mathcal{F}_{MSG}^{(l)}\left(\boldsymbol{h}_v^{(l-1)}, \boldsymbol{h}_u^{(l-1)}\right), \forall (u,v) \in E \tag{5}$$

$$\boldsymbol{a}_v^{(l)} = \mathcal{F}_{AGG}^{(l)}\left(\left\{\boldsymbol{m}_{uv}^{(l)} | u \in N(v)\right\}\right), \forall v \in V \tag{6}$$

$$\boldsymbol{h}_v^{(l)} = \mathcal{F}_{UPT}^{(l)}\left(\boldsymbol{h}_v^{(l-1)}, \boldsymbol{a}_v^{(l)}\right), \forall v \in V \tag{7}$$

where the operations $\mathcal{F}_{MSG}$, $\mathcal{F}_{AGG}$, and $\mathcal{F}_{UPT}$ are parameterized functions with subscripts denoting "message," "aggregate," and "update," respectively, and $\boldsymbol{h}_v^{(l)}$ is the node representation (or embedding vector) at the lth layer. Concretely, as shown in Figure 4, all the nodes in the graph carry a "message" (initially, the message is the original node feature), and at every layer, each node aggregates

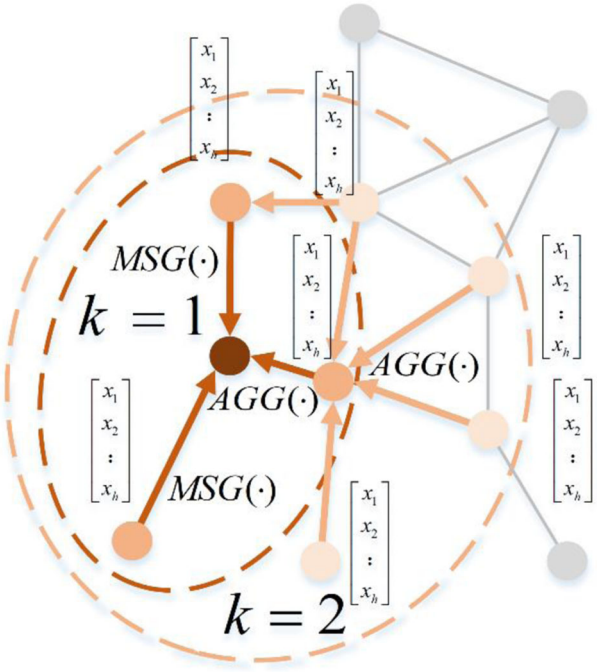


FIGURE 4 Message passing mechanism of GNNs

"messages" from their neighborhood nodes to update the next message or embedding. At the output head, GNNs implement the specified task, such as node classification and link prediction, based on the final representation $\boldsymbol{h}_v^{(l)}$(Wu et al., 2020).

Typically, $\mathcal{F}_{MSG}^{(l)}$ is embedded into the aggregation function $\mathcal{F}_{AGG}^{(l)}$ or simply preserves the representation from the previous layer. The choices of $\mathcal{F}_{AGG}^{(l)}$ and $\mathcal{F}_{UPT}^{(l)}$ are crucial and derive different GNN models. For instance, the graph convolutional network (Kipf & Welling, 2017) selects mean pooling as the aggregation function and a single-layer neural network as the updater:

$$\boldsymbol{h}_v^{(l)} = \mathrm{ReLU}\left(\boldsymbol{W}^{(l)} \cdot \mathrm{MEAN}\left\{\boldsymbol{h}_u^{(l-1)} | \forall u \in N(v) \cup \{v\}\right\}\right) \tag{8}$$

Alternatively, in GraphSAGE (Hamilton et al., 2017), a max pooling aggregator is used, and the updater adds a concatenation operation before the linear mapping:

$$\boldsymbol{h}_v^{(l)} = \mathrm{ReLU}\left(\boldsymbol{W}_{UPT}^{(l)} \cdot \left[\boldsymbol{h}_v^{(l-1)}, \boldsymbol{a}_v^{(l)}\right]\right) \tag{9}$$

$$\boldsymbol{a}_v^{(l)} = \mathrm{MAX}\left(\left\{\mathrm{ReLU}\left(\boldsymbol{W}_{AGG}^{(l)} \cdot \boldsymbol{h}_u^{(l-1)}\right), \forall u \in N(v)\right\}\right) \tag{10}$$

The lReLU operation in Equations (8) to (10) is the rectified linear unit:

$$\mathrm{ReLU}(x) = x \odot 1_{x \geq 0} \tag{11}$$

where $\mathbf{1}_{condition}$ is the indicator function and $\odot$ denotes the Hadamard product.

3.2 | Modification of the GIN

In the structural analysis scenario, an effective GNN should be able to distinguish different nodes by mapping them to different representations in the embedding space. Figure 5 displays an asymmetric frame structure, wherein nodes i and j have the same degree (i.e., the number of edges connecting to the node). However, despite sharing the same local structure, the internal forces at these two nodes react differently in most load cases. Moreover, since these two nodes are located in the middle of the entire structure, very deep layers are required to reach the boundary so that the model can discriminate the subtrees rooted in them, which is infeasible in practice. Therefore, how GNNs can identify different nodes with similar local structures is a challenging problem in structural analysis and imposes high demands on selecting an appropriate aggregation function within the message-passing framework.

To address this problem, we first augment the graph data by introducing physical coordinates into the node features. In this manner, nodes that have the same local structure, such as node i and node j in Figure 5, can be distinguished easily because the messages (i.e., the coordinates) passed from their neighborhood nodes differ and thus can be projected to discriminative representations through simple mathematical operations. This method corresponds to a feature augmentation strategy in which node IDs are used to enhance the expressiveness of graph data, which is popular in recommender systems (J. Wang et al., 2018).

For nodes with different degrees, such as nodes s and j, we resort to carefully designing the message aggregator. We revisit classical aggregators—sum pooling, average pooling, and max pooling—to analyze their effectiveness. On the right side of Figure 5, we illustrate the "messages" from the neighborhood of nodes s and j. We calculate the aggregation results using classical aggregators as shown in Table 1. By comparing the results at two nodes, we can determine the cases in which sum pooling and mean pooling fail to distinguish the local structures. For example, when $a = 7d$, $b = 7\,h$, $x = 9d$, and $y = 9\,h$, the sum pooling aggregator yields the same message information of *(36d, 36 h)*. Max pooling seems the most powerful in this situation. However, in Figure 5, nodes s and r, which share the same maximal neighborhood coordinate ($a + d$, $b + h$), will be confused. In summary, classical aggregators are incapable of adapting to all cases in structural systems.

Distinguishing different nodes with similar local structures corresponds to the well-known graph isomorphism problem (Weisfeiler & Leman, 1968). An eligible GNN is expected to map different multisets of node features to different representations, requiring the aggregation function to be injective. Considering that the neighborhood nodes in structural systems are countable and finite, we adopt the philosophy of GIN (Xu et al., 2019), which states that for a countable multiset, there exists a function $\mathcal{G} : \mathcal{X} \to \mathbb{R}^n$ so that

$$\mathcal{H}(v, X) = (1 + \epsilon) \cdot \mathcal{G}(v) + \sum_{x \in X} \mathcal{G}(x) \tag{12}$$

is unique, where $v \in \mathcal{X}$ and $X \subset \mathcal{X}$ is a multiset of bounded size. In addition, any function $\mathcal{F}$ that acts on such pairs can be decomposed as follows:

$$\mathcal{F}(v, X) = \phi\left[(1 + \epsilon) \cdot \mathcal{G}(v) + \sum_{x \in X} \mathcal{G}(x)\right] \tag{13}$$

Based on this statement, GIN uses multilayer perceptrons (MLPs) to model the composite function $\mathcal{G}^{(l)} \circ \phi^{(l-1)}$ by leveraging the universal approximation theorem (Hornik et al., 1989, 1990). The original GIN thus updates node representations as follows:

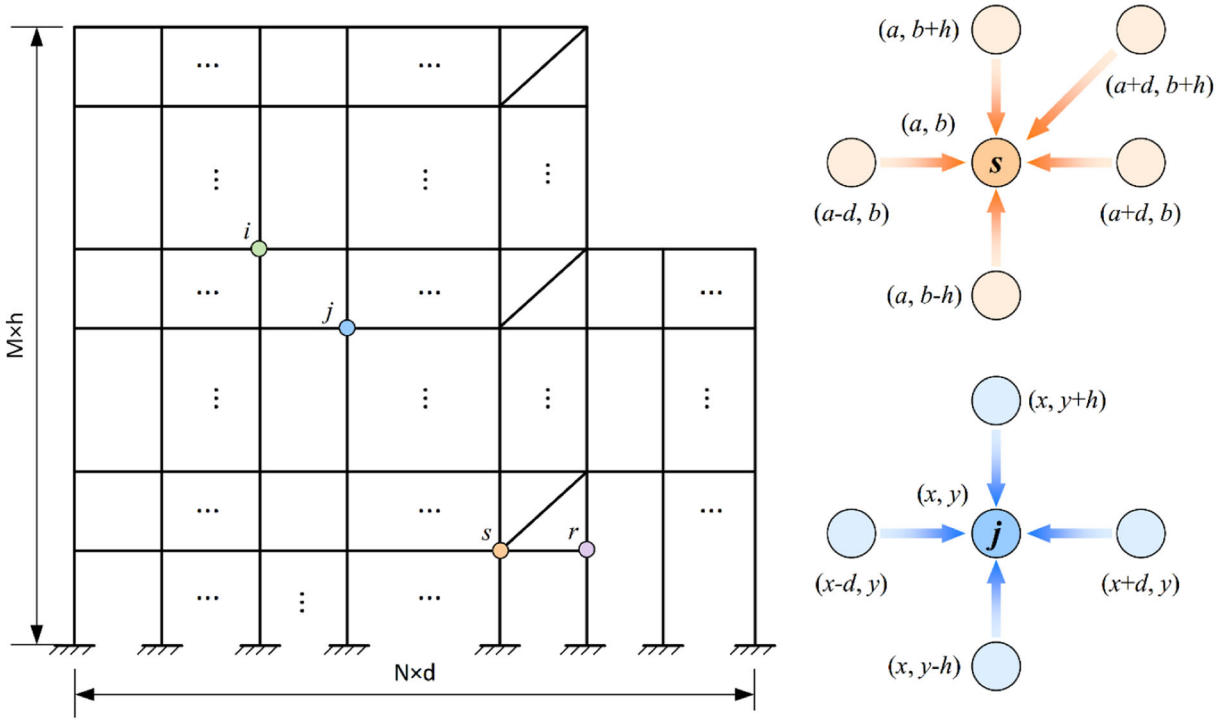


FIGURE 5 Illustration of message passing in a frame structure

TABLE 1 Failure cases of classical aggregators

	Sum pooling	Mean pooling	Max pooling
Node s	$(5a + d, 5b + h)$	$(a + d/5, b + h/5)$	$(a + d, b + h)$
Node j	$(4x, 4y)$	(x, y)	$(x + d, y + h)$
Failure case	$\begin{cases} 5a + d = 4x \\ 5b + h = 4y \end{cases}$	$\begin{cases} 5a + d = 5x \\ 5b + h = 5y \end{cases}$	–

$$\boldsymbol{h}_v^{(l)} = \mathrm{MLP}^{(l)}\left(\left(1 + \epsilon^{(l)}\right) \cdot \boldsymbol{h}_v^{(l-1)} + \sum_{u \in N(v)} \boldsymbol{h}_u^{(l-1)}\right) \quad (14)$$

where $\epsilon^{(l)}$ can be either learnable or fixed. Equation (12) is applicable when node features are organized as one-hot encodings in the first iteration, which makes their summation injective. To accommodate structural system scenarios, we modify the first layer to:

$$\boldsymbol{h}_v^{(1)} = \mathrm{MLP}^{(1)}\left(\left(1 + \epsilon^{(1)}\right) \cdot \mathrm{MLP}^{(0)}\boldsymbol{x}_v + \sum_{u \in N(v)} \mathrm{MLP}^{(0)}\boldsymbol{x}_u\right) \quad (15)$$

Then, the GIN model can preserve injectiveness while recursively updating each node's feature embedding to capture the graph structure.

3.3 | Physics-informed modeling

The mainstream DL models are driven by big data, mining the latent patterns underlying the dataset. Accordingly, existing ML/DL studies on structural analysis rely heavily on data from experiments or generated using FE analysis to train the model parameters, which corresponds to a supervised learning scheme. However, in contrast to construction materials and structural members such as beams and columns, experimental data are very scarce at the structural system level. Meanwhile, data generation through FE analysis will consume a large amount of time and, more importantly, cannot cover the entire parameter space due to the high complexity of feature information of structural systems. In addition to the data paucity problem, data-driven models are criticized for their poor interpretability and neglect the physical background of structural engineering. As a result, researchers and engineers cannot evaluate the correctness of the prediction results, which impedes the promotion of DL methods in engineering projects.

Structural analysis has a solid theory in mechanics. Specifically, elastic cases are governed by three mechanical equations: the equilibrium equation, the deformation compatibility equation, and the elastic constitutive

equation. According to structural mechanics, these three equations are theoretically complete, implying that if they are satisfied everywhere in an elastic structural system, the solutions of internal forces and deformation are correct and unique. Based on this concept, we innovatively employ a self-supervised learning scheme to free the model from the demand in data and propose a new training mode driven by structural mechanics.

Analogous to classical solutions in structural mechanics, we focus on the deformation and internal forces at member nodes. By virtue of graph representation, we can maintain node degrees of freedom (DOFs) as extra node features and easily realize the deformation compatibility equation because it is convenient to condense the DOFs according to rigid or hinge connections. The elastic constitutive relationship is stored in the extended adjacency matrix or edge features. Therefore, the DL model only needs to address equilibrium equations, from which we can construct a mechanistic-based loss function for optimization:

$$\mathcal{L} = \|\boldsymbol{F}_{\text{in}} + \boldsymbol{F}_{\text{ex}}\|, \tag{16}$$

where $\boldsymbol{F}_{\text{in}}$ denotes the internal force vectors computed by the model, $\boldsymbol{F}_{\text{ex}}$ denotes the external input load vectors, and $\|\cdot\|$ is a norm metric, such as the mean squared error (MSE) loss with the L2-norm. Accordingly, solving equilibrium equations is equivalent to minimizing Equation (16). In contrast to the "train-test-generalize" paradigm of data-driven models, the convergence of the training process of our physics-informed model corresponds to the completion of the solution, whose correctness is guaranteed by the theoretical completeness of the three mechanical equations.

Figure 6 displays the algorithm of the StructGNN-E model in force-driven cases, where the external loads are the inputs, and the model attempts to predict the node deformation according to the structural topological information and node (as well as edge) features. The entire procedure of the physics-informed model no longer requires labeled data (processed data labeled with prediction targets by humans, often used for training and validation of data-driven models), which fundamentally alleviates the problem of the severe data paucity of structural systems and poses no restrictions on the structures to be simulated, leading to great generalization. Moreover, the resolution mainly relies on the topological messages from the underlying representation, giving full play to the advantages of the graph data structure, which can be interpreted as automatic identification of the force transmission path inside the structural system through DL.

To further illustrate the process, we demonstrate the setup of an example of StructGNN-E with a 2-hop GIN:

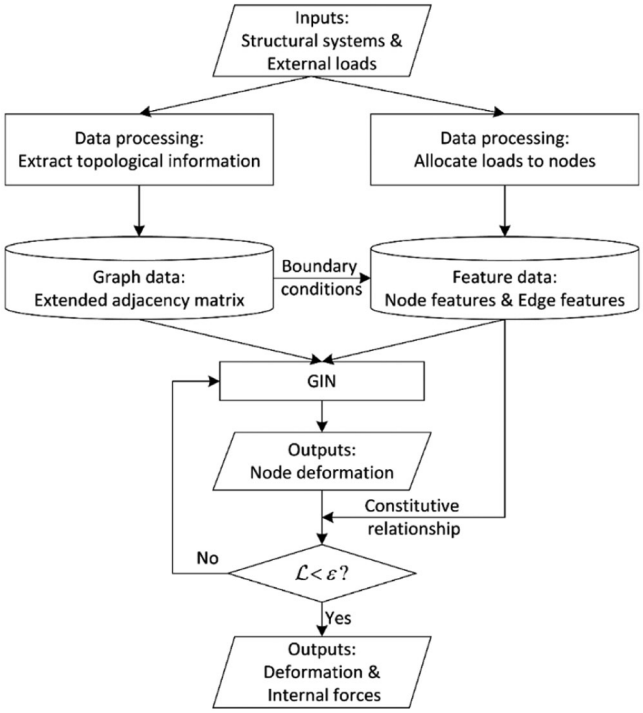


FIGURE 6 StructGNN-E model algorithm in force-driven cases. GIN, graph isomorphism network

The input I_{v} includes node coordinates (x_v, y_v) and nodal load $\boldsymbol{F_{ex}}$:

$$I_v = (x_v, y_v, F_{vx}, F_{vy}, M_{vy}) \tag{17}$$

The output O_{v} is set as the global nodal displacement:

$$O_v = (dx_v, dy_v, d\theta_v) \tag{18}$$

With a 2-hop GIN, the input is processed as follows:

$$h_v^{(0)} = Linear_1(I_v) \tag{19}$$

$$h_v^{(1)} = \text{ReLU}[MLP^{(1)}(h_v^{(0)} + \sum_n A_{v,u} h_u^{(0)})] \tag{20}$$

$$h_v^{(2)} = MLP^{(2)}(h_v^{(1)} + \sum_n A_{v,u} h_u^{(1)}) \tag{21}$$

$$O_v = Linear_2(h_v^{(2)}) = (dx_v, dy_v, d\theta_v) \tag{22}$$

$\boldsymbol{F_{in}} = (F_x, F_y, M)$ is then computed by the output with constitutive relationships of structure members.

Thus far, we have elaborated on the mechanism of the StructGNN-E model, which is capable of recognizing

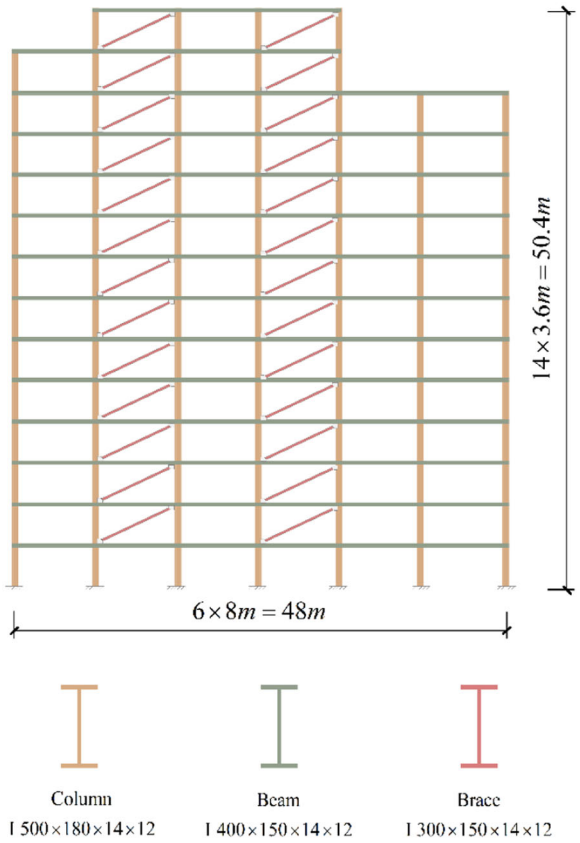


FIGURE 7 A frame structure generated for the numerical experiment

different node representations with similar local topologies and performing data-free structural analysis driven by structural mechanics.

4 | VALIDATION OF THE STRUCTGNN-E MODEL

To validate the effectiveness and efficiency of the StructGNN-E model for elastic structural analysis quickly and informatively, we carry out a numerical experiment using multifloor frame structures with diverse configurations. We compare the results with numerical results produced by the FE method to show its correctness.

4.1 | Data preparation

To prepare the data for the numerical experiment, we develop a subroutine that can randomly generate steel frame structures with braces conforming to the common engineering design. The number of floors and spans and the member sections can be manipulated in the subroutine to vary the configurations of the generated structures. For example, Figure 7 shows a 14-floor five-span steel frame structure with braces generated by the subroutine, wherein the floor height is 3.6 m and the span length is 8 m. For simplicity, all columns have an I-section of 500 × 180 × 14 × 12 ($h_w \times b_f \times t_f \times t_w$ mm), all beams have an I-section of 400 × 150 × 14 × 12 (mm), and all braces have an I-section of 300 × 150 × 14 × 12 (mm). All structural members adopt Q345B structural steel with an elastic modulus of 200 GPa. External loads are also generated randomly, including node forces and moments. Intuitively, the number of nodes can reflect the scale of a structural system, which also indicates the size of the GNN model. We validate our proposed model on different-scale problems with 10, 100, and 1000 nodes.

TABLE 2 Number of parameters for different problem scales (number of nodes *N*)

Problem scale (*N*)	First layer dimension	Second layer dimension	#Trainable parameters (*P*)
10	8	32	3411
100	16	64	12,867
1000	32	128	49,472

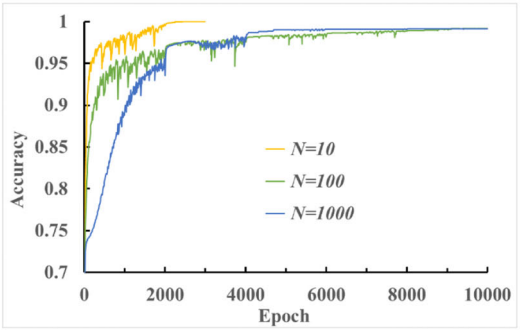
4.2 | Training configurations

The StructGNN-E model can convert the generated structures into graph data (including an extended adjacency matrix and node and edge features) and build a GIN model with a proper dimension. We adopt a two-layer GIN model that covers two hops with two-layer MLPs. The dimensions of the hidden states and the total number of trainable parameters for different problem scales are listed in Table 2. Although the learnable parameters increase as the problem scales up, the size of our model is still much smaller than that of common DL models.

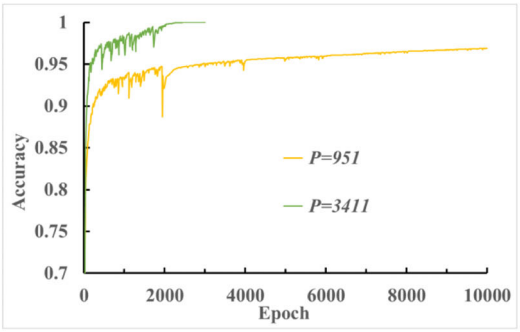
The model is trained using the Adam algorithm (Kingma & Ba, 2014) with default settings of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ and employs a learning rate schedule that decays the initial learning rate ($lr_0 = 0.01$) with epochs. The MSE loss is adopted to measure the residual forces in Equation (16). For each problem scale (number of nodes *N*), 100 cases of different loading conditions, including the lateral load and vertical weight (randomly generated), are calculated, each of which stops when the number of epochs reaches 10,000 or the relative residual forces are less than 1%. The StructGNN-E model is developed based on the PyTorch platform and is run on an i7-8700 CPU and an RTX Titan GPU (graphic processing unit) to test the efficiency.

4.3 | Model validation

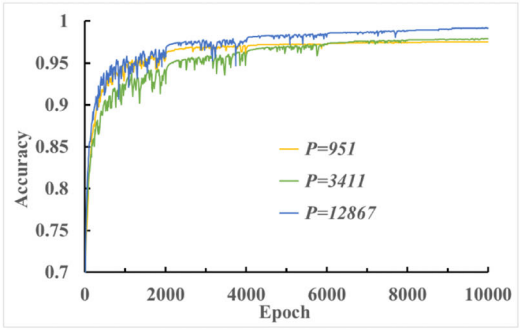
As shown in Figure 8a, the StructGNN-E model converges in all cases and presents reasonable results. The accuracy



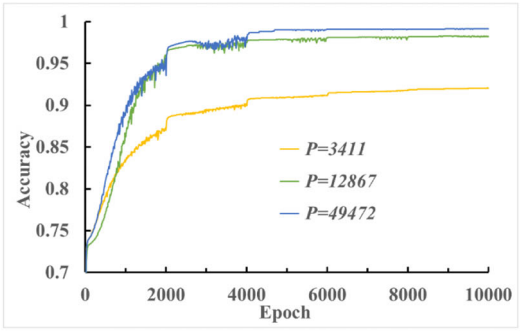
(a) Average convergence curves for different problem scales.



(b) Comparison of convergence curves of different model sizes on problem N=10.



(c) Comparison of convergence curves of different model sizes on problem N=100.



(d) Comparison of convergence curves of different model sizes on problem N=1000.

FIGURE 8 Average convergence curves of different model sizes on different problem scales. (a) Average convergence curves for different problem scales, (b) comparison of convergence curves of different model sizes on problem $N = 10$, (c) comparison of convergence curves of different model sizes on problem $N = 100$, and (d) comparison of convergence curves of different model sizes on problem $N = 1000$

TABLE 3 Comparison of the computational efficiency

Problem scale (N)	SAP2000	StructGNN-E (CPU)	StructGNN-E (GPU)
10	0.15 s	0.61 s	0.42 s
100	3.20 s	3.36 s	1.94 s
1000	68.27 s	71.94 s	45.69 s
10,000	2185.20 s	1390.63 s	848.54 s

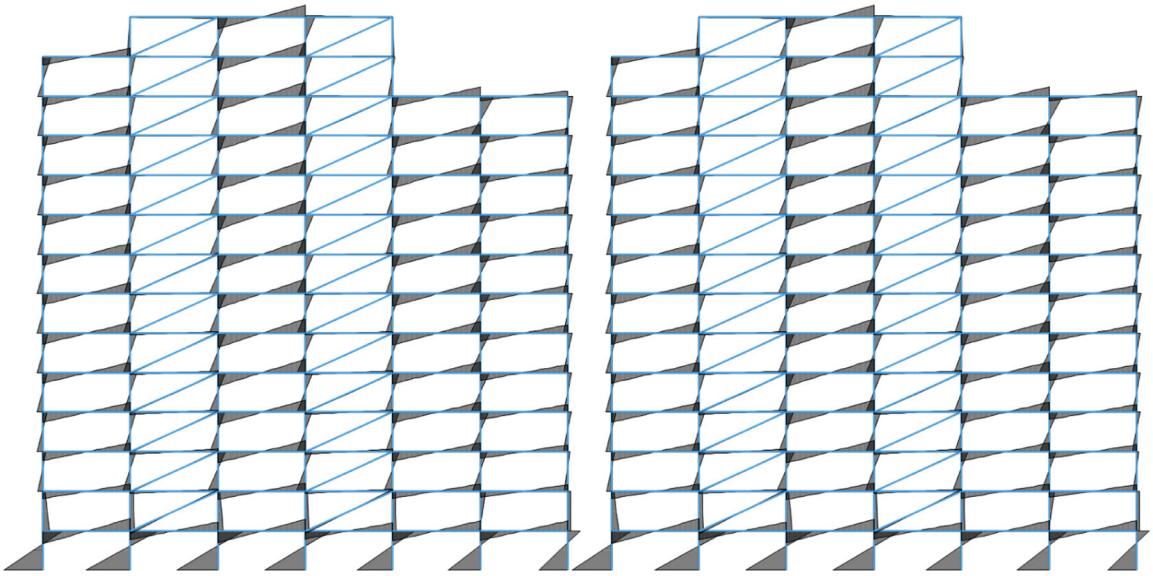
is defined according to Equation (16) as

$$Accu = 1 - \|\boldsymbol{F}_{\text{in}} + \boldsymbol{F}_{\text{ex}}\| / \|\boldsymbol{F}_{\text{ex}}\| \tag{23}$$

The StructGNN-E model achieves an average accuracy of 99% on different problem scales with random loading conditions, indicating that our proposed model and theory-driven paradigm are exceedingly effective and are capable of accurately implementing elastic structural analysis. Figure 8b–d shows the influences of the number of learnable parameters P on the performance, showing that with the increase in problem scale, the model also needs to expand its size to adapt to higher complexity. Although a too small model size P may limit the convergence, quantitative analysis using logarithmic transformation indicates that the suitable model size ensuring convergence is approximately $C{\cdot}N^{0.58}$ in terms of the problem scale N (number of nodes), where C is a constant. For the same problem scale, larger models yield higher accuracy and slightly faster convergence.

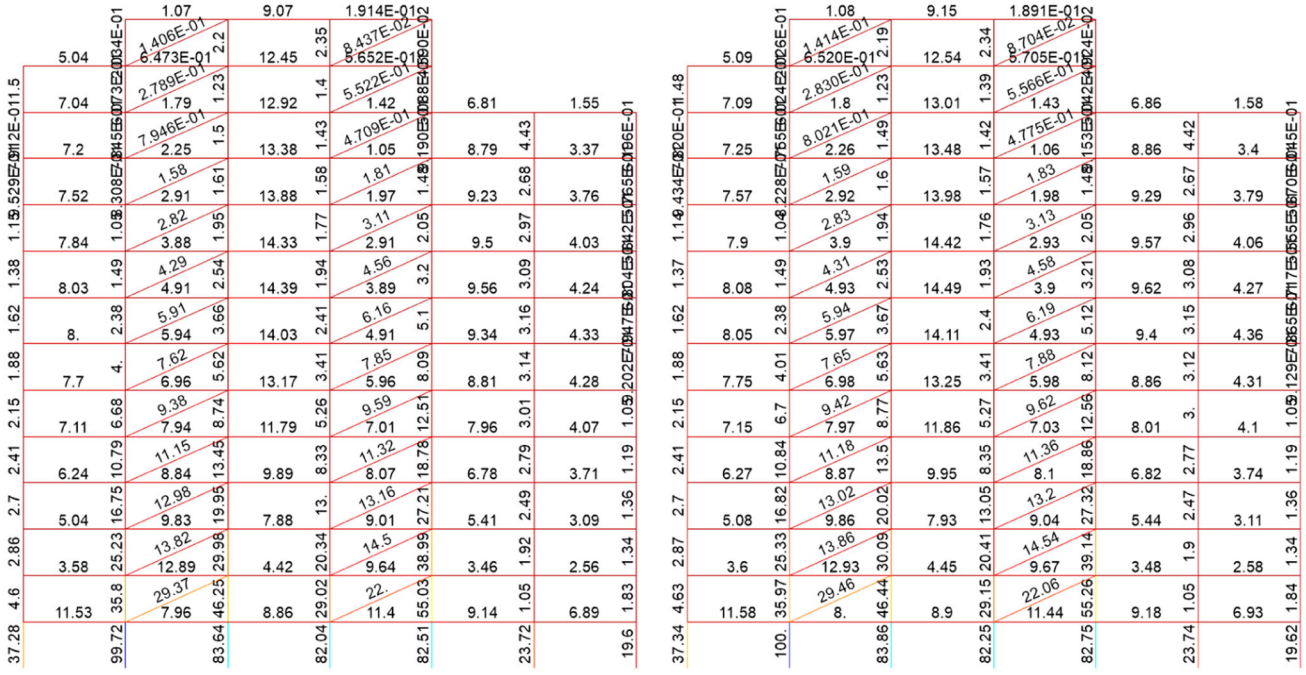
To show the performance more clearly, we compare the StructGNN-E model with the commonly used FE package SAP2000. The analysis results are displayed in Figures 9 and 10, where both the bending moment diagram and the virtual work diagram computed by the model agree very well with the FE model, verifying the high accuracy of the StructGNN-E model. Table 3 shows a comparison of the computational efficiency of the different models. Generally, the models are run on GPUs to utilize their optimized computing power. However, to ensure comparability with the FE software, we add a column to show the computational efficiency of the StructGNN-E model run on one CPU. Table 3 indicates that our proposed model exhibits competitive efficiency on the CPU, compared with the FE software and even surpasses the latter when $N = 10{,}000$, saving up to 36% of the time cost. The model run on the GPU is much faster than expected, which verifies that the StructGNN-E model is efficient for implementing structural analysis, especially when the problem scale is large.

In summary, the numerical experiment verifies the effectiveness of the StructGNN-E model, which can solve the elastic analysis problem of structural systems of any scale with high accuracy and excellent computational



(a) Results using the StructGNN-E model **(b)** Results using SAP2000

FIGURE 9 Comparison of bending moment diagrams. (a) Results using the StructGNN-E model and (b) results using SAP2000



(a) Results using the StructGNN-E model **(b)** Results using SAP2000

FIGURE 10 Comparison of virtual work diagrams. (a) Results using the StructGNN-E model and (b) Results using SAP2000

efficiency, making our proposed model applicable to engineering practice.

5 | DISCUSSION

This section presents ablation studies to further demonstrate the unique effectiveness of the StructGNN-E model at the structural system level and conceptually discuss its potential application in the field of structural optimization.

5.1 | Analysis between GNN and classical DL models

The GNN architecture developed on the graph data structure is the skeleton of the StructGNN-E model.

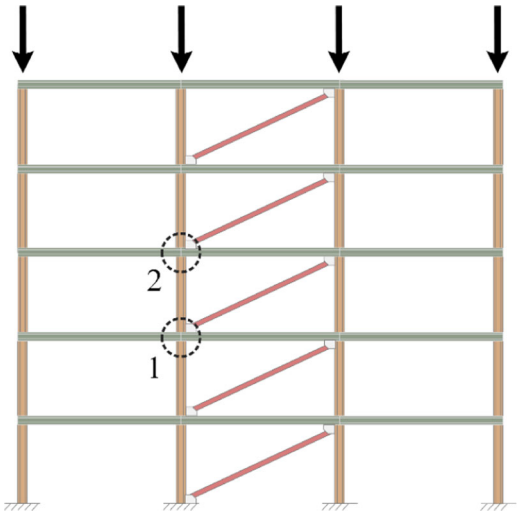


FIGURE 11 Two similar nodes without external loads in a frame structure

Accordingly, we first compare the GNN architecture with classical DL models, including a feedforward deep neural network (DNN), CNN, and RNN.

To accommodate classical DL models, the data inputs need to be modified. Intuitively, the structural analysis seeks to build a mapping relation between two mechanical variables (C. Wang et al., 2020), such as the load and displacement. However, as shown in Figure 11, many nodes have the same local topologies and the same input loads (or no load) in a structure, resulting in the same inputs into deep learning models such as a DNN or CNN. Since the classical DL models are deterministic, they can only yield the same results on these nodes, which obviously fails to predict correct displacements. An RNN is an exception because it can complement inputs by receiving historical states $\boldsymbol{h}_i^{(l)}$ from the previous unit (shown in Figure 12), helping to distinguish the nodes to some extent. Therefore, we use node coordinates as the inputs to the DNN and CNN, whereas we use external loads as the inputs to the RNN. All the outputs are nodal displacements in Equation (18). The process remains the same as shown in Figure 6 except for the GIN part.

We experiment on two structures with $N = 100$ and 1000. In each case, the learnable parameters in each model are kept similar, as listed in Table 4. The models all adopt a physics-informed mode. Other experimental configurations remain the same, including the optimization algorithm and the learning rate scheme.

The comparison results are presented in Figure 13 and Table 4, wherein only the StructGNN-E model manages to converge to acceptable accuracy. For the CNN, as seen in Figure 14, the convolutional kernel maps the coordinates of "neighborhood nodes" to new representations. Such a "neighborhood" relationship is determined by data

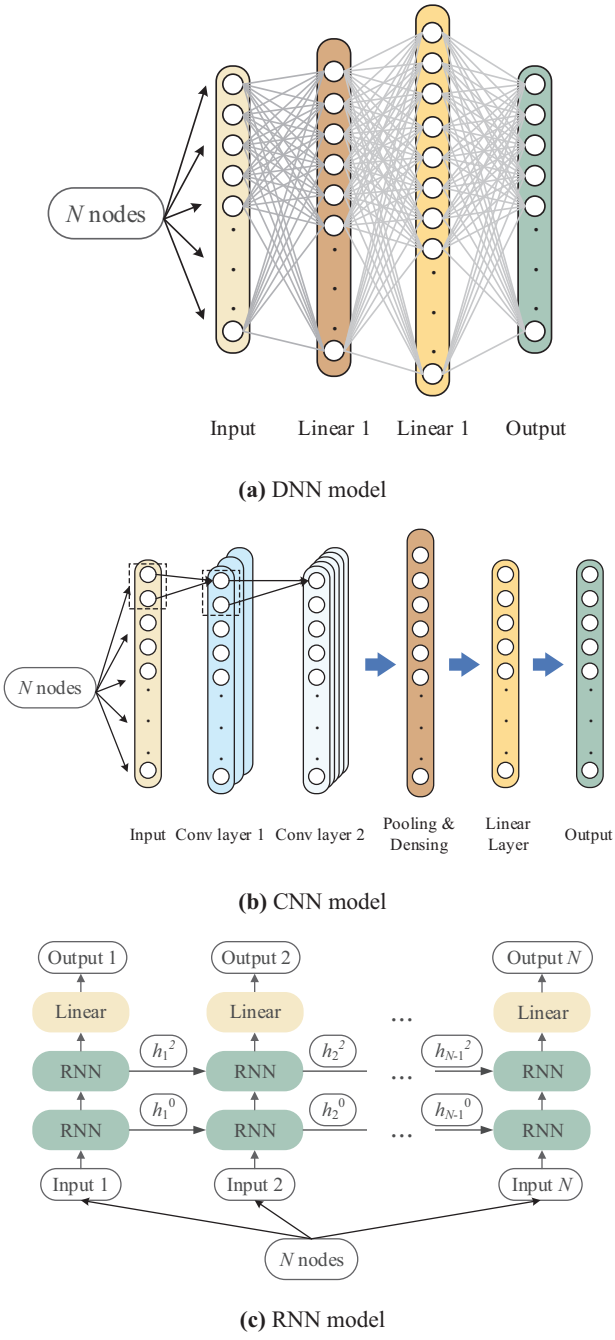
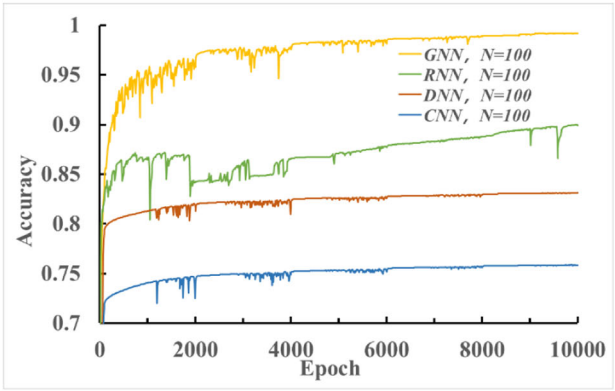
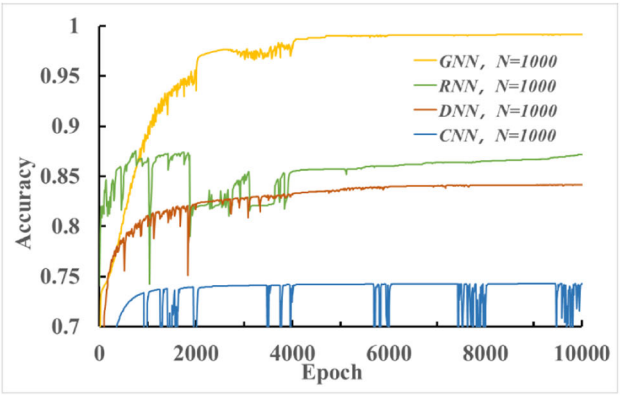


FIGURE 12 DNN/convolutional neural network (CNN)/RNN model architectures. (a) DNN model. (b) CNN model. (c) RNN model

organization. For example, Nodes 2 and 3 are adjacent in the input data, but they actually have no direct mechanical interaction in the structural system. Consequently, the local information extracted by the CNN is useless for the final solution. In contrast to the CNN, the DNN intends to capture all information by full neural connections (as shown in Figure 14), thus achieving better performance. However, full connections also introduce many irrelevant signals that disturb the model as it attempts to find the



(a) Comparison of convergence curves of different models on problem N=100.



(b) Comparison of convergence curves of different models on problem N=1000.

FIGURE 13 Average convergence curves of different models on different problem scales. (a) Comparison of convergence curves of different models on problem $N = 100$ and (b) comparison of convergence curves of different models on problem $N = 1000$

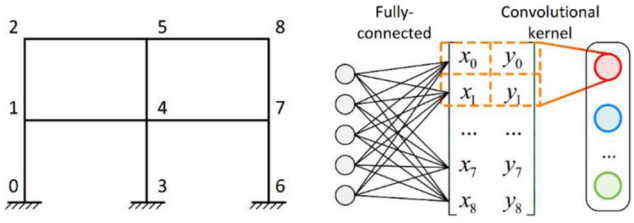


FIGURE 14 Illustration of the DNN and CNN operating on a frame structure

optimal solution, resulting in model divergence. Augmented by the historical states, the RNN relies on the information transmitted from the previous unit with an underlying assumption that the nodes in the structure have order. As a result, if we permute the node IDs, the RNN will predict different deformations, which is inconsistent with the actual situation.

In summary, this comparison experiment again reveals the nonsequential and translation-variant properties of structural systems and demonstrates the infeasibility of classical DL models in handling structural systems. On the other hand, the graph data structure and the GNN architecture developed on it succeed in capturing the distinctive characteristics of structural systems and exhibit unique validity.

5.2 | Analysis between data-driven modeling and physics-informed modeling

Another focal point in the StructGNN-E model is the physics-informed paradigm, which resolves the data scarcity problem and ensures the theoretical correctness of the computational results. To discuss the superiority of physics-informed modeling over data-driven modeling at the structural system level, we use SAP2000 to generate the load-displacement data of two structures with $N = 100$ and 1000 for training the supervised learning model. We prepare a dataset with a total size of 1000 (1000 different loading conditions randomly generated) for each case and divide 800 pieces of data into the training dataset and the rest into the test dataset. In each case, the graph representation and the GIN model of the same dimension are adopted. For the physics-informed paradigm, the convergence of the training process of our physics-informed model corresponds to the completion of the solution, which means that no labeled data or division of training/test is necessary. Other experimental configurations also remain the same for comparability.

The results are presented in Figure 15 and Table 5. Figure 15 indicates that the data-driven model converges on the training dataset, realizing accuracies of 99.2% and 99.1% on the two structures. However, on the test dataset,

TABLE 4 Comparison of the StructGNN-E model and classical deep learning models

Scale (N)		DNN	Convolutional neural network	RNN	GNN
100	#Parameters	4739	3171	3699	3411
	Accuracy	83.1%	89.9%	75.8%	99.2%
1000	#Parameters	17,667	12,099	15,603	12,867
	Accuracy	84.2%	87.6%	74.3%	99.1%

TABLE 5 Comparison of data-driven modeling and physics-informed modeling

Scale (N)		Data-driven modeling	Physics-informed modeling
100	Training accuracy	99.2%	–
	Test accuracy	74.6%	99.2%
1000	Training accuracy	99.0%	–
	Test accuracy	73.2%	99.1%

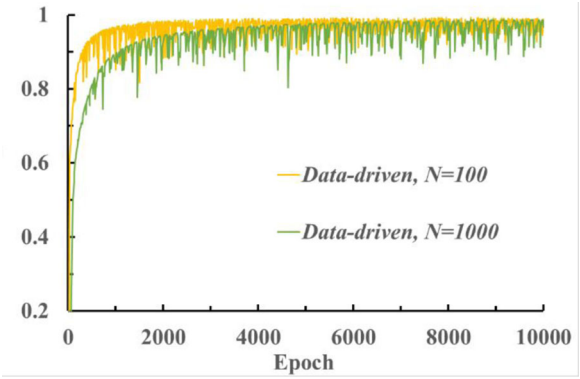


FIGURE 15 Accuracy of the data-driven model on the training dataset

the data-driven model exhibits very poor generalization and only achieves accuracies of 74.6% and 73.2%, far lower than those on the training dataset. In contrast, the physics-informed model achieves superb performance on the test dataset, demonstrating its effectiveness compared with the data-driven counterpart. We speculate that the failure of the data-driven model is ascribed to the large hypothesis space at the structural system level. For a small-scale planar frame structure with 100 nodes, even considering only three types of node loads (i.e., x-direction force F_x, y-direction force F_y, and moment M), there are $2^3 = 8$ possibilities of load conditions at each node, resulting in 8^{100} load conditions for the entire structure. The displacement space also has a similar scale of possibilities. Therefore, the hypothesis space of mapping from the load space to the displacement space (or vice versa) can be very large and too difficult for the data-driven method to learn a well-generalized model with limited data. The physics-informed model overcomes this difficulty by imposing strict physical constraints derived from structural mechanics on the hypothesis space, wherein the equilibrium equation is deemed a strong regularization that prunes the searching processes to a large extent. Accordingly, the physics-informed method is able to reach the optimal solution more quickly and easily.

In summary, the data-driven paradigm is inapplicable to structural systems, whereas the physics-informed model exhibits unique effectiveness.

5.3 | Potential application in end-to-end structural optimization

We have detailed the advantages of StructGNN-E in the above analysis. In addition to its unique effectiveness and excellent computational accuracy and efficiency, it is worth noting that the StructGNN-E model is capable of propagating differentiability due to the backpropagation algorithm in DL, which has great potential to reform the structural optimization procedures.

Structural optimization is a popular problem in the engineering design and construction stages, wherein engineers attempt to adjust the structural configurations according to engineering indices, such as a balanced combination of safety and economy. Traditionally, structural optimization was implemented manually by iteratively varying the input parameters (such as structural sections) and checking whether the results were acceptable. Consequently, it was very difficult to acquire a real optimal solution with a compromise in engineering applications. In recent decades, researchers have been dedicated to investigating automatic computer algorithms as a substitute for manual labor. However, obstructed by conventional computational methods represented by FE analysis (shown in the upper part of Figure 16), the influences of the input parameters on the target indices cannot be traced back, especially when the coupling effect exists in the input space. Due to this restriction, heuristic algorithms such as the genetic algorithm (Gholizadeh et al., 2008; Jenkins, 1991; Liu et al., 2008) were widely adopted by researchers because they were model-agnostic and automatically mutated the encodings of the input parameters to minimize the engineering targets, wherein only the forward direction was needed so that the entire procedure conformed to the unidirectional paradigm of traditional methods. These classical algorithms are inefficient because the results provide a bare minimum of guidance for searching optimal parameters, and the variation of parameters is more or less random. In addition, the parameterization of traditional structural analysis models is poor. Changes in input parameters usually mean building a new model from scratch, which hinders the realization of automatic optimization. Therefore, classical algorithms have not been promoted for use in structural optimization problems.

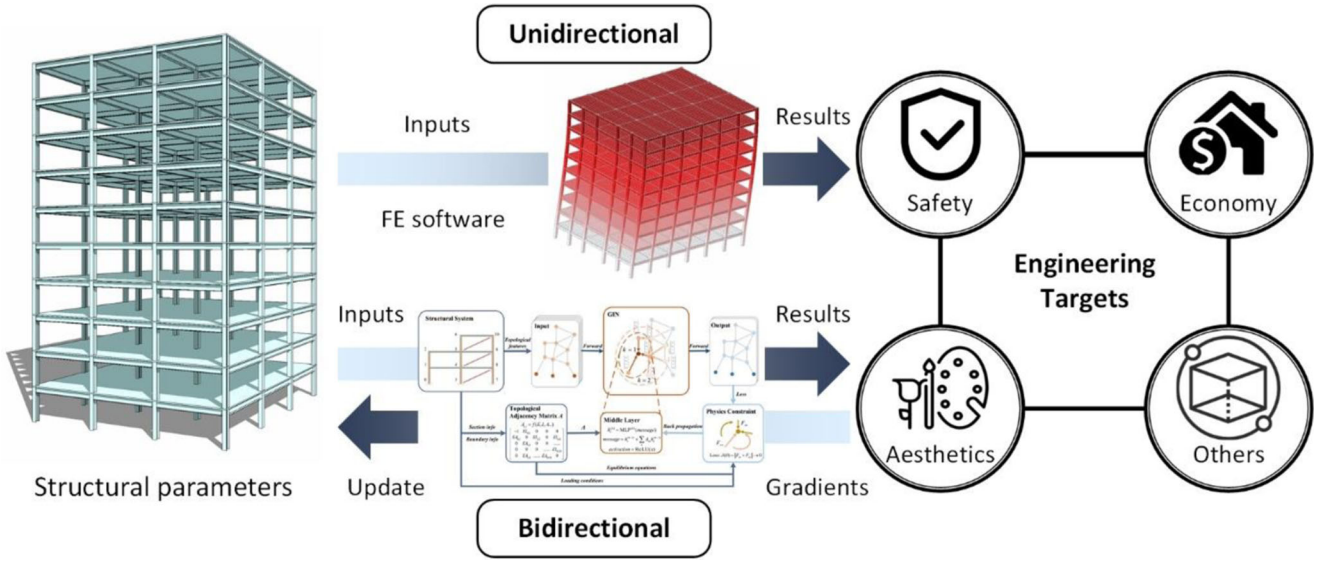


FIGURE 16 Structural optimization procedures with traditional methods and deep learning methods. FE, finite element

In contrast, the StructGNN-E model can modify the traditional unidirectional paradigm to a new bidirectional paradigm. As shown in Figure 16, not only can the input parameters be transformed into the final results, but the gradients of the results can also be propagated back to update the input parameters. The backward flow is implemented by assimilating the input parameters as a part of learnable parameters in DL, which can be co-optimized by the gradient descent method. Due to the excellent parameterization of DL, there is no need to rebuild the model when the input parameters are changed. This new paradigm is much more efficient because, on the one hand, the information from the results is fully utilized to guide the optimization direction of the input parameters, and on the other hand, the optimizer integrated into the DL framework, such as the Adam algorithm, is more advanced than traditional algorithms. Moreover, since the optimization and the structural analysis share the same training process, we can accomplish them simultaneously and not avoid implementing two separate algorithms. Therefore, the StructGNN-E model possesses great potential to innovate the field of structural optimization, deriving a new end-to-end optimization paradigm with high efficiency and automaticity.

6 | CONCLUSION

To fill the research gap of ML/DL-based analysis at the structural system level, we propose a physics-informed DL model named StructGNN-E based on the graph data structure. A numerical experiment and ablation studies demonstrate that our proposed model is uniquely effective, compared with classical DL models and exhibits superb performance in both computational accuracy and efficiency.

The main conclusions are summarized as follows:

1. We innovatively utilize the graph data structure to organize the feature information with nonsequential and translation-variant properties, which realizes the data representation of structural systems with high fidelity.
2. We propose the StructGNN-E model within the GNN architecture to adapt to the graph representation, wherein we modify GIN to distinguish the nodes with similar local topologies.
3. We propose a novel physics-informed paradigm to resolve the data scarcity problem at the structural system level. The paradigm incorporates structural mechanics into DL and converts the training process to solving the physical equations, implementing the structural analysis without labeled data and ensuring the theoretical correctness.
4. The numerical experiment verifies that StructGNN-E converges to accurate results of structures with different scales and exhibits excellent computational efficiency.
5. Ablation studies demonstrate the unique effectiveness of StructGNN-E at the structural system level compared with classical DL models and the data-driven paradigm.
6. The StructGNN-E model can derive a new end-to-end structural optimization method with high efficiency and automaticity, which is anticipated to have great potential to reform the field of structural optimization.

Nevertheless, the current framework is applicable for elastic analysis and frame-like structures. The implementation of the physics-informed paradigm can be complex

for elasto-plastic scenarios when the memory effect is taken into account. Structures with high-dimensional members (such as shear walls) or nonlinear deformation are difficult to fit into the current framework. Despite these limitations, StructGNN-E provides a valuable technical framework for intelligent computation at the structural system level, including a graph data representation to describe the structural topologies, a GNN architecture to gather the information from neighborhood nodes, and a physics-informed paradigm to alleviate the dependence on big data, which paves an inspiring avenue for our future work on extending it to elasto-plastic structural analysis.

ACKNOWLEDGMENTS

We gratefully acknowledge the financial support provided by the National Natural Science Foundation of China (Grant No. 52121005), China National Postdoctoral Program for Innovative Talents (Award No. BX20220177), and China Postdoctoral Science Foundation (Grant No. 2022M711864).

REFERENCES

Amezquita-Sancheza, J. P., Valtierra-Rodriguez, M., & Adeli, H. (2020). Machine learning in structural engineering. *Scientia Iranica*, *27*(6), 2645–2656.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *240*, 100379.

Esteghamati, M. Z., & Flint, M. M. (2021). Developing data-driven surrogate models for holistic performance-based assessment of mid-rise RC frame buildings at early design. *Engineering Structures*, *245*, 112971.

Feng, D. C., Liu, Z. T., Wang, X. D., Chen, Y., Chang, J. Q., Wei, D. F., & Jiang, Z. M. (2020). Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials*, *230*, 117000.

Gholizadeh, S., Salajegheh, E., & Torkzadeh, P. (2008). Structural optimization with frequency constraints by genetic algorithm using wavelet radial basis function neural network. *Journal of Sound and Vibration*, *312*(1-2), 316–331.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press.

Graf, W., Freitag, S., Sickert, J. U., & Kaliske, M. (2012). Structural analysis with fuzzy data and neural network based material description. *Computer-Aided Civil and Infrastructure Engineering*, *27*(9), 640–654.

Guan, X., Burton, H., Shokrabadi, M., & Yi, Z. (2021). Seismic drift demand estimation for steel moment frame buildings: From mechanics-based to data-driven models. *journal of structural engineering*, *147*(6), 04021058.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, *30*, 1025.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, *3*(5), 551–560.

Huang, P., & Chen, Z. (2021). Deep learning for nonlinear seismic responses prediction of subway station. *Engineering Structures*, *244*, 112735.

Hung, S., & Jan, J. (2002). Machine learning in engineering analysis and design: An integrated fuzzy neural network learning mode. *Computer-Aided Civil and Infrastructure Engineering*, *14*(3), 207–219.

Hwang, S. H., Mangalathu, S., Shin, J., & Jeon, J. S. (2021). Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *Journal of Building Engineering*, *34*, 101905.

Jenkins, W. M. (1991). Towards structural optimization via the genetic algorithm. *Computers & Structures*, *40*(5), 1321–1327.

Kabir, M. A. B., Hasan, A. S., & Billah, A. H M. (2021). Failure mode identification of column base plate connection using data-driven machine learning techniques. *Engineering Structures*, *240*, 112389.

Kang, M. C., Yoo, D. Y., & Gupta, R. (2021). Machine learning-based prediction for compressive and flexural strengths of steel fiber-reinforced concrete. *Construction and Building Materials*, *266*, 121117.

Kangwai, R. D., & Guest, S. D. (2000). Symmetry-adapted equilibrium matrices. *International Journal of Solids and Structures*, *37*(11), 1525–1548.

Kim, T., Kwon, O. S., & Song, J. (2019). Response prediction of nonlinear hysteretic systems by deep neural networks. *Neural Networks*, *111*, 1–10.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *The 3rd International Conference on Learning Representations*, San Diego, CA.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, Toulon, France.

Lagaros, N. D., & Papadrakakis, M. (2012). Neural network based prediction schemes of the non-linear seismic response of 3D buildings. *Advances in Engineering Software*, *44*(1), 92–115.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Lee, S., Ha, J., Zokhirova, M., Moon, H., & Lee, J. (2018). Background information of deep learning for structural engineering. *Archives of Computational Methods in Engineering*, *25*(1), 121–129.

Liu, X., Yi, W. J., Li, Q. S., & Shen, P. S. (2008). Genetic evolutionary structural optimization. *Journal of Constructional Steel Research*, *64*(3), 305–311.

Lu, J., Luo, Y., & Li, N. (2009). An incremental algorithm to trace the non-linear equilibrium paths of pin-jointed structures using the singular value decomposition of the equilibrium matrix. *Proceedings of the Institution of Mechanical Engineers*, *223*(7), 881–890.

Martins, G. B., Papa, J. P., & Adeli, H. (2020). Deep learning techniques for recommender systems based on collaborative filtering. *Expert Systems*, *37*(6), e12647.

Messner, J. I., Sanvido, V. E., & Kumara, S. R. (1994). StructNet: a neural network for structural system selection. *Computer-Aided Civil and Infrastructure Engineering*, *9*(2), 109–118.

Morfidis, K., & Kostinakis, K. (2017). Seismic parameters' combinations for the optimum prediction of the damage state of RC buildings using neural networks. *Advances in Engineering Software*, *106*, 1–16.

Naderpour, H., Mirrashid, M., & Parsa, P. (2021). Failure mode prediction of reinforced concrete columns using machine learning methods. *Engineering Structures*, *248*, 113263.

Naser, M. Z. (2021). An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: Navigating causality, forced goodness, and the false perception of inference. *Automation in Construction*, *129*, 103821.

Nguyen, T., Kashani, A., Ngo, T., & Bordas, S. (2018). Deep neural network with high-order neuron for the prediction of foamed concrete strength. *Computer-Aided Civil and Infrastructure Engineering*, *34*(4), 316–332.

Oh, B. K., Park, Y., & Park, H. S. (2020). Seismic response prediction method for building structures using convolutional neural network. *Structural Control and Health Monitoring*, *27*(5), e2519.

Olalusi, O. B., & Awoyera, P. O. (2021). Shear capacity prediction of slender reinforced concrete structures with steel fibers using machine learning. *Engineering Structures*, *227*, 111470.

Parvin, A., & Serpen, G. (1999). Recurrent neural networks for structural optimization. *Computer-Aided Civil and Infrastructure Engineering*, *14*(6), 445–451.

Pellegrino, S. (1993). Structural computations with the singular value decomposition of the equilibrium matrix. *International Journal of Solids and Structures*, *30*(21), 3025–3035.

Pellegrino, S., & Calladine, C. R. (1986). Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures*, *22*(4), 409–428.

Rafiei, M. H., & Adeli, H. (2017). NEEWS: A novel earthquake early warning system using neural dynamic classification and neural dynamic optimization model. *Soil Dynamics and Earthquake Engineering*, *100*, 417–427.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete compressive strength. *ACI Materials Journal*, *114*(2), 237–244.

Reddy, J. N. (2019). *Introduction to the finite element method* (4th ed.). McGraw-Hill Education.

Salehi, H., & Burgueño, R. (2018). Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, *171*, 170–189.

Sun, H., Burton, H. V., & Huang, H. (2021). Machine learning applications for building structural design and performance assessment, state-of-the-art review. *Journal of Building Engineering*, *33*, 101816.

Torky, A. A., & Ohno, S. (2021). Deep learning techniques for predicting nonlinear multi-component seismic responses of structural buildings. *Computers & Structures*, *252*, 106570.

Tsubaki, M., Tomii, K., & Sese, J. (2019). Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, *35*(2), 309–318.

Wakjira, T. G., Al-Hamrani, A., Ebead, U., & Alnahhal, W. (2022). Shear capacity prediction of FRP-RC beams using single and ensenble ExPlainable machine learning models. *Composite Structures*, *287*, 115381.

Wang, C., Song, L., & Fan, J. (2022). End-to-end structural analysis in civil engineering based on deep learning. *Automation in Construction*, *138*, 104255.

Wang, C., Xu, L., & Fan, J. (2020). A general deep learning framework for history-dependent response prediction based on UA-Seq2Seq model. *Computer Methods in Applied Mechanics and Engineering*, *372*, 113357.

Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., & Lee, D. L. (2018). Billion-scale commodity embedding for e-commerce recommendation in Alibaba. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK (pp. 839–848).

Wang, Y., Wang, J., Cao, Z., & Farimani, A. B. (2022). Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, *4*, 279–287.

Weisfeiler, B., & Leman, A. (1968). A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, *2*(9), 12–16.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2019). How powerful are graph neural networks? *International Conference on Learning Representations*, New Orleans, LA.

Yin, T., & Zhu, H. P. (2019). An efficient algorithm for architecture design of Bayesian neural network in structural model updating. *Computer-Aided Civil and Infrastructure Engineering*, *35*(4), 354–372.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK (pp. 974–983).

Zafeiriou, S., Bronstein, M., Cohen, T., Vinyals, O., Leskovec, J., Liò, P., Bruna, J., & Gori, M. (2022). Guest editorial: Non-euclidean machine learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *44*(2), 723–726.

Zhang, R., Chen, Z., Chen, S., Zheng, J., Büyüköztürk, O., & Sun, H. (2019). Deep long short-term memory networks for nonlinear structural seismic response prediction. *Computers & Structures*, *220*, 55–68.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81.

Zhu, S., Ohsaki, M., & Guo, X. (2021). Prediction of non-linear buckling load of imperfect reticulated shell using modified consistent imperfection and machine learning. *Engineering Structures*, *226*, 111374.

Zienkiewicz, O. C., & Robert, L. T. (2005). *The finite element method for solid and structural mechanics*. Elsevier.

How to cite this article: Song, L.-H., Wang, C., Fan, J.-S., & Lu, H.-M. (2023). Elastic structural analysis based on graph neural network without labeled data. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 1307–1323. https://doi.org/10.1111/mice.12944