



Research
Civil Engineering—Article

Efficient Full-Range Nonlinear Analyses of Structural Systems Based on Heterogeneous Graph Learning



Linghan Song^{a,b}, Chen Wang^{b,*}, Jiansheng Fan^{b,c}

^a College of Civil Engineering, Fuzhou University, Fuzhou 350108, China

^b Department of Civil Engineering, Tsinghua University, Beijing 100084, China

^c Key Laboratory of Civil Engineering Safety and Durability of the Ministry of Education, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:

Received 14 July 2024

Revised 14 April 2025

Accepted 1 July 2025

Available online 7 July 2025

Keywords:

Structural system analysis
Heterogeneous graph-based deep learning
Full-range response prediction
Digital twin
Smart structural engineering

ABSTRACT

Nonlinear analyses possess tremendous significance throughout the entire lifespans of civil structures. In recent years, the interest in leveraging deep learning (DL) to address the efficiency limitations of the traditional structural analysis methods has increased. However, full-range nonlinear analyses of different structures remain underresearched because of a lack of appropriate data representations and the failure to consider both internal structural information and external load conditions. A heterogeneous graph (HetG) representation scheme that can digitalize arbitrary structural systems with high fidelity is proposed in this study. Furthermore, a composite feature learning framework is developed to enable efficient full-range nonlinear analyses. This framework comprises two main components: ① a heterogeneous graph neural network (GNN)-based module that encodes static features into embeddings with full structural semantics and ② a sequence-to-sequence (Seq2Seq) module that predicts history-dependent responses using structural embeddings and external stimuli in an end-to-end manner. A computational model named structural analysis based on a graph neural network-nonlinear (StructGNN-N) is implemented based on the proposed methodology and is validated through numerical experiments involving real-world concrete structures. The results show that StructGNN-N successfully reproduces the full-range nonlinear responses of all nodes in the entire structure and exhibits excellent generalizability across structures with diverse topological designs and member configurations. Notably, the developed model achieves a computational efficiency level that is 1000 times greater than that of the traditional elastoplastic history analysis approach using the finite-element (FE) method. A parametric analysis and ablation studies demonstrate the effectiveness of the StructGNN-N architecture. Due to its superior accuracy and computational efficiency, the proposed method holds great potential for use in engineering applications, especially in the context of digital twins. This approach provides an inspiring path for simulating diverse engineering structures with accurate and comprehensive mechanical information in real time.

© 2025 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Nonlinear structural analyses form a fundamental aspect of structural engineering throughout the entire lifespan of a structure, from elastoplastic time history analysis during the design phase to reliability estimation of damaged structures during maintenance. This technique plays a pivotal role in evaluating the safety and functionality of buildings and infrastructures while also link-

ing physical structures to their digitalized representations [1,2]. The conventional methods, which are primarily based on the finite-element (FE) framework, face challenges such as high modeling complexity, low computational efficiency, and convergence issues [3]. These limitations make the traditional methods increasingly inadequate for modern engineering applications [4], especially within the digital twin context [5], which demands real-time updates and simulations.

In recent years, the structural analysis technique leveraging new-generation artificial intelligence (AI), represented by deep learning (DL) [6–8], has gained significant attention due to its extraordinary performance [4], greatly reducing the required

* Corresponding author.

E-mail address: chwang@tsinghua.edu.cn (C. Wang).

analysis time while maintaining superior accuracy. Furthermore, AI-based structural analyses exhibit impressive adaptability, enabling a single well-trained model to handle diverse scenarios without repetitive model construction processes [4,9]. Notably, with the evolution of the AI for Science (AI4S) paradigm, DL models can efficiently uncover the intrinsic physical properties within complex mechanical problems, even when the underlying mechanisms are unclear [10,11]. These strengths have motivated numerous scholars to integrate DL into nonlinear structural analysis methods.

The current research can be categorized into two levels: analyses of individual structural components and analyses of structural systems. At the structural component level, Tao et al. [12] employed deep neural networks (DNNs) for the constitutive modeling of composites, achieving considerable accuracy and demonstrating the effectiveness of neural networks in mechanical analysis scenarios. Rahman et al. [13] compared various machine learning models for estimating the shear strength of steel-fiber reinforced concrete (RC) beams and suggested suitable formations for structural engineers. Abambres and Lantsoght [14] built a neural network to predict the shear capacities of one-way slabs under concentrated loads, demonstrating its powerful ability to simulate complex mechanical behaviors. Feng et al. [15] used a data-driven approach to train models for predicting the plastic hinge lengths of RC columns, achieving significantly higher prediction accuracy than that of the state-of-the-art empirical relationship-based methods. To conduct structural system analyses, Huang et al. [16] developed a DL-based model for predicting the seismic responses of a subway station, demonstrating that DL methods can be used in basic nonlinear analyses. Ning et al. [17] combined long short-term memory (LSTM), convolutional neural networks (CNNs), and WaveNet to analyze the seismic responses of a steel building frame and bridge structures; this was a successful attempt to integrate the characteristics of DL methods with nonlinear analyses of structural systems. However, the applicability of this technique is relatively restricted to simplified structural systems because of the limitations of the underlying data scheme. Zhou et al. [18] employed a physics-informed DL model for predicting the real-time responses of a high-rise building and achieved high accuracy by incorporating physical information into the model. Fei et al. [19] proposed a data-driven prediction method for evaluating shear wall structures, which were simplified to multiple degrees of freedom (MDOF) systems.

Despite these substantial research advancements, a significant gap remains between academic studies and the practical application of the state-of-the-art AI models for analyzing real engineering structures. The primary limitation is the inability of such models to compute full-range responses for an entire structure. The current methodologies predominantly focus on surrogate modeling, typically predicting specific mechanical indices at limited points of interest, such as the displacement history of the top floor of a building [4]. In contrast, the classic FE method can obtain the holistic mechanical fields of structures. This limitation stems from the lack of appropriate data schemes that are capable of digitalizing arbitrary structural systems with high fidelity. Notably, system-level analysis studies lag behind component-level analysis studies because of their intricate nature, which is characterized by diverse spatial topologies and complex component configurations. Conventional linear data structures (e.g., vectors, sequences, and grids) fall short in terms of capturing the comprehensive features of engineering structures. Consequently, the existing models either address a specific structural system without structural information [18] or oversimplify regular structures into finite features, thereby failing to provide global mechanistic response predictions for the entire target structure. In previous research [20], the authors proposed a physics-informed graph neural network (GNN) for elastic structural analyses, which introduced graph data—a type of non-

Euclidean data structure—to parameterize frame structures. This innovation could be extended to represent arbitrary structural systems. However, the proposed model was unable to process nonlinear stimuli and was restricted to linear elasticity. A full-range nonlinear analysis of a structural system necessitates a methodology that considers both the inherent features of the underlying structure and the complete time history of the associated external loading cases.

To address the aforementioned gap, a heterogeneous graph (HetG) learning-based computational model, structural analysis based on a graph neural network-nonlinear (StructGNN-N), is proposed in this study to conduct full-range nonlinear analyses of structural systems. The structure of this study is as follows. Section 2 presents a universal graph data representation scheme for structural systems and elucidates the corresponding computational framework. Section 3 presents the methodology. Section 4 details the implementation of the proposed methodology, including a GNN-transformer model and a novel strategy for alleviating the data requirements of training. Section 5 validates the model, while Section 6 discusses the results of an ablation study and explores potential applications involving digital twins. Finally, Section 7 concludes our research.

2. Related work

Heterogeneous graph neural networks (HGNNs). Research on HGNNs has gained significant traction over the past decade because of their effectiveness in managing HetG data [21,22]. Unlike traditional methods that encode information derived from uniform nodes and edge types, HetG learning techniques have been designed to preserve the relational diversity present in real-world data, encompassing various entities and connections. For example, Hu et al. [23] proposed heterogeneous graph transformer (HGT) and employed an attention mechanism to facilitate message passing among different types of nodes. Mo et al. [24] introduced a novel framework, heterogeneous edge-enhanced graph attention network (HEAT), that effectively models the heterogeneous dynamics and interactions of multiagent data using directed edge-feature graphs and adaptive map selection. To mitigate the performance degradation that occurs in deeper HGNNs, metapaths have been incorporated into models to connect distant nodes, leading to improved models such as HGNN-attribute enhancement and structure-aware attention (AESA) [25] and metapath context convolution-based heterogeneous graph (MECCH) [26]. The versatility of HGNNs has led to their widespread application in areas such as social networks and recommendation systems. Cai et al. [27] utilized a HetG encoder network, heterogeneous graph contrastive learning (HGCL), for learning the representations of users and microvideos, along with a type-crossing objective function that integrates embeddings from diverse node types. Another HGCL framework presented by Chen et al. [28] enhances the adaptive contrastive learning process conducted between user-item interactions and auxiliary heterogeneous relational views, thereby more effectively capturing diverse user preferences and item characteristics. Ma et al. [29] designed heterogeneous generative pre-trained transformer (HetGPT), which introduces a multiview neighborhood aggregation mechanism to capture the complex neighborhood structures that are inherent in HetGs and improves applications involving complex patterns such as online classification and social recommendation. The research and applications related to HGNNs inspire the foundational framework presented in this paper, which seeks to integrate an approach for managing different types of nodes into the representation learning processes of structural systems, thereby enabling the heterogeneous components within these systems to be more accurately modeled.

GNN-based mechanical analysis. In recent years, intuitive methods involving graph data and GNNs, which represent physics problems with node and edge configurations, have been applied in the field of solid mechanics, including metamaterials and composite materials [30]. Black and Najafi [31] employed a multifidelity GNN as a supervised framework for solving two-dimensional (2D) elastoplastic problems concerning metamaterials. Li et al. [32] proposed a machine learning approach based on a GNN to simulate the dynamic responses of copper bars and demonstrated the general applicability of their GNN in structural response prediction tasks. Xue et al. [33] designed a graph-based network to solve dynamic problems related to mechanical metamaterials; their approach can incorporate defects and spatial inhomogeneities. Yacouti and Shakiba [34] combined a CNN and a GNN to predict the mechanical fields contained in composite microstructures and discussed the advantages of each network component. Hendriks et al. [35] designed a similarity-equivariant GNN for the homogenization of metamaterials, which innovatively introduces physical symmetries to predict mechanical properties. Further studies involving GNN applications have addressed various problems, including mesh-based simulations of continua [36], dominant failure mode searches for structures [37], and dynamic response predictions for continuous deformable bodies [38], offering new perspectives regarding structural mechanics. Inspired by the related studies and the similarities between structural systems and graph data, the previous research published by the authors in Ref. [20] was among the first works to propose the use of homogeneous GNNs for representing and computationally analyzing structural systems, and effectively addressed elastic analyses of frame structures. Zhang et al. [39,40] represented architectural and structural layouts using graph data and developed a computational and optimization framework for RC systems by integrating GNNs and generative adversarial networks (GANs). Following the same scheme, Zhao et al. [41] discussed the possibility of extracting topological information from the layouts of shear wall systems, and Fei et al. [42] improved the computational efficiency achieved in the design and construction stages with knowledge-enhanced GNNs. In addition, Zhang et al. [43] used a specifically tailored GNN with adaptive capabilities to conduct seismic safety assessments across various scenarios involving coupled train–bridge systems. These studies have preliminarily demonstrated the potential of GNNs for simulating the elastic behaviors of structural systems via message passing mechanisms; however, few studies have focused on elastoplastic analyses, which are pervasive in engineering applications. Furthermore, the existing studies have typically concentrated on a single structural system or a specific category, simplifying the internal entities to a single type and overlooking the mechanical differences among various component types. As a result, the employed models are predominantly homogenous GNNs. To produce more accurate representations of general structural systems composed of multiple entity types, it is necessary to incorporate an HetG for data representation and computation purposes.

3. Methodology

3.1. Graph representations of arbitrary structural systems

Before delving into DL modeling, it is crucial to establish high-fidelity representation schemes for diverse structural systems on the data side. Our previous research [20] laid the groundwork by transforming bar systems, such as frame structures with beams and columns, into graph data structures [44,45] (Fig. 1(a)), thus effectively mapping joints to nodes and structural bars to edges.

However, many engineering structures encompass plate and shell components, such as shear wall buildings and nuclear containments. To accommodate this broader spectrum of scenarios, we propose a novel approach using HetGs [22] representations for arbitrary structural systems.

In the new scheme, different structural components and joints (i.e., the intersections of components) correspond to different types of nodes. The features of the structural components and joints (e.g., their section geometries, material properties, and joint coordinates), which are denoted as static features [46], are organized within the node features. Since the descriptions of different structural components vary, a node in the HetG can carry heterogeneous content. Furthermore, the connectivity among distinct graph nodes inherently represents the topological relationships contained within the structural system.

Mathematically, considering a structural system comprising M_0 structural joints and k types of structural components (each with corresponding counts of M_1, M_2, \dots, M_k). The node feature (v_i) of a node i that belongs to a certain type of node is as follows:

$$v_i = (v_{i1}, v_{i2}, \dots, id_i) \quad (1)$$

where d_i denotes the feature dimensionality of node i . The topology of the structural system is represented by an adjacent matrix (\mathbf{A}) with a size of $N \times N$ (N is the total number of nodes):

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}, N = \sum_k M_k \quad (2)$$

where each element a_{ij} denotes the connectivity between nodes i and j . If $a_{ij} = 1$, then the nodes i and j are connected. Notably, \mathbf{A} is symmetric.

On this basis, we can digitalize any structure system using an HetG:

$$G = \{V, E, \mathbf{A}, \psi\} \quad (3)$$

where G is the graph data, V is the set of nodes, and E is the set of edges; each node belongs to a set of nodes with the same category ψ . Conversely, given reasonable HetG data, we can uniquely determine the corresponding structural system. In this manner, a bijective mapping relationship between an arbitrary structural system and the HetG is established.

A simplified frame–shear wall structure is taken as an example (Fig. 1(b)). Its HetG representation encompasses 30 nodes categorized into six types, with four columns, eight beams, two braces, two shear walls, two slabs, and 12 joints (including four column feet). Each node type is visually differentiated using distinct colors. Moreover, assuming that the node features include joint coordinates, employing the breadth–first search algorithm facilitates a straightforward process of reconstructing the initial structure.

Building upon the new scheme, we propose the following DL modeling pipeline (Fig. 2) to address the problem of full-range nonlinear structural analyses. Utilizing the HetG representation, an HGNN framework is employed to encode the static features of each component within the structural system, serving as structural embeddings. The input stimulus sequences, denoted as dynamic features [46,47], are subsequently integrated with the structural embeddings and channeled through a sequence processing network to generate full-range responses.

3.2. Structural embedding with an HGNN

The raw features of a structural system are recorded in the HetG representation but are not suitable for direct computation because

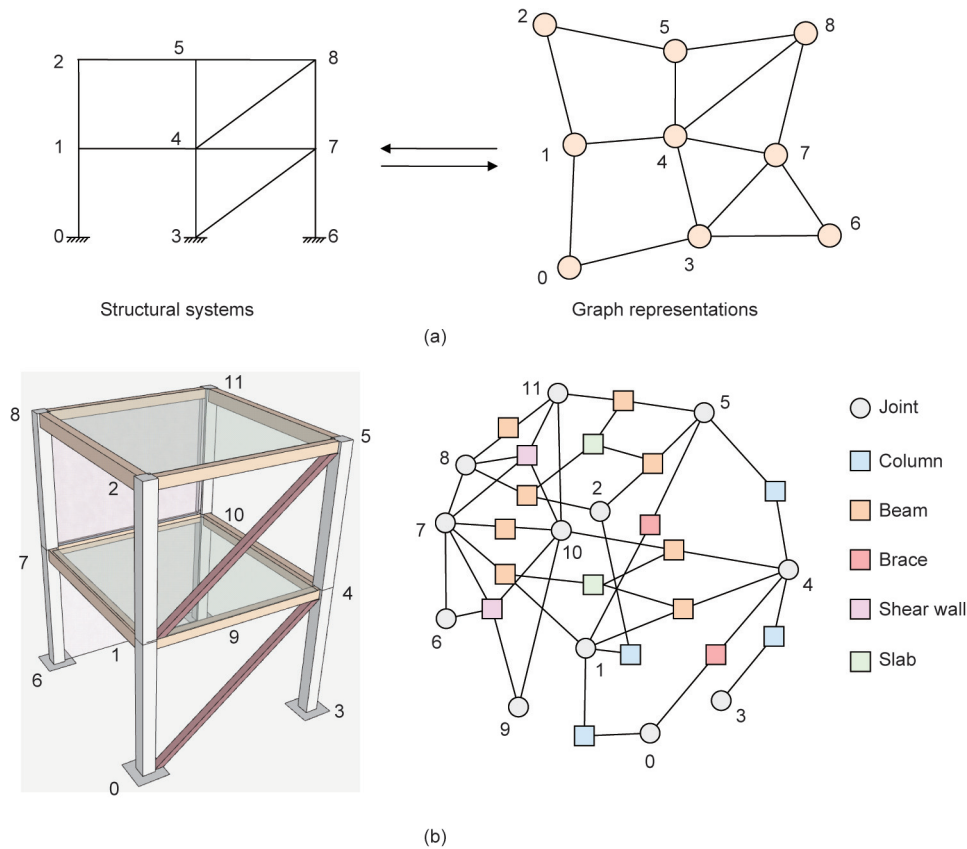


Fig. 1. Schematic examples of structural systems and their homogenous HetG representations. (a) A schematic example of a structural system and its homogenous graph representations, where the numbers denote the joint nodes; (b) a schematic example of a structural system and its HetG representations.

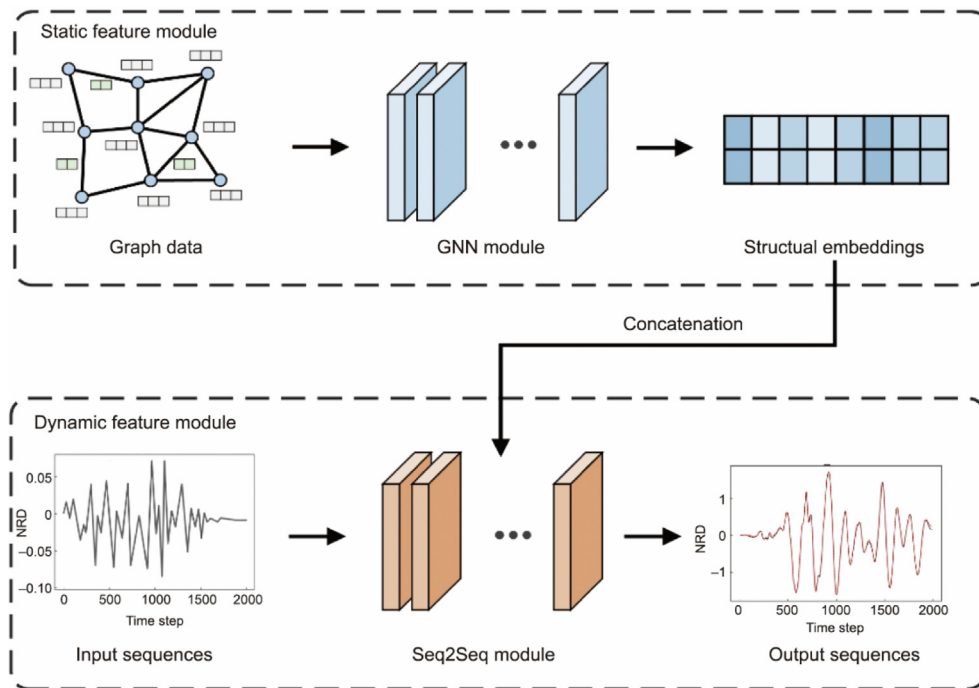


Fig. 2. Model pipeline. Seq2Seq: sequence-to-sequence. NRD: normalized displacement.

the features of distinct structural components are isolated and topological information is not integrated. To learn the collaborative mechanism used by the structural components within the system, the HGNN is utilized for processing various types of graph nodes.

GNNs belong to a category of DL models that are designed to capture the nonlinear relationships embedded in graph data [44]. Following the methodology introduced in Ref. [20], the learning process of the HGNN can adopt the message passing mechanism:

$$\mathbf{h}_i^{(0)} = \mathcal{T}_{\psi(\mathbf{v}_i)}(\mathbf{v}_i), \forall \mathbf{v}_i \in V \quad (4)$$

$$\mathbf{m}_{ij}^{(l)} = \mathcal{M}^{(l)}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}), \forall (\mathbf{v}_i, \mathbf{v}_j) \in E \quad (5)$$

$$\mathbf{a}_i^{(l)} = \mathcal{G}^{(l)}\left(\left\{\mathbf{m}_{ij}^{(l)} \mid \forall j \neq i \& a_{ij} = 1\right\}\right), \forall \mathbf{v}_i \in V \quad (6)$$

$$\mathbf{h}_i^{(l)} = \mathcal{U}^{(l)}(\mathbf{h}_i^{(l-1)}, \mathbf{a}_i^{(l)}), \forall \mathbf{v}_i \in V \quad (7)$$

where $\mathcal{T}_{\psi(\mathbf{v}_i)}$ is an operator that is used to map different types of nodes to a common latent space; $\mathcal{M}^{(l)}$, $\mathcal{G}^{(l)}$, and $\mathcal{U}^{(l)}$ are the learnable functions denoting the node message interconnection, aggregation, and update operations in the l th layer, respectively; \mathbf{v}_j is the node feature of the node j ; $\mathbf{m}_{ij}^{(l)}$ is the message from nodes j to i in the l th layer; $\mathbf{h}_i^{(l)}$ and $\mathbf{A}_i^{(l)}$ are the state information and aggregated information of the node i in the l th layer, respectively; $\mathbf{h}_i^{(l-1)}$ and $\mathbf{h}_j^{(l-1)}$ represent the state information of nodes i and j in the $(l-1)$ th layer, respectively. Through the iterative message passing scheme, the HGNN can fuse topological and structural component configuration information into the node representations, yielding structural embeddings with rich mechanistic semantics.

3.3. Structural response generation

The primary objective of the full-range nonlinear analysis process is to predict response sequences under various loading protocols. In principle, a time series-based DL model can address tasks involving sequence data [48,49]. However, classic computational mechanics reveal a pronounced history-dependent effect in structural nonlinear analysis scenarios, necessitating intricate evolution rules to formulate hysteretic behaviors. For example, steel plate shear walls, which are some of the most common laterally resistant components in engineering structures, exhibit significant cyclic hardening and breathing effects [47]. Consequently, distinct loading paths can yield various energy dissipation performances [50].

To accommodate the demand for history-dependent sequential prediction, the sequence-to-sequence (Seq2Seq) [51] architecture integrated with an attention mechanism [52] (Fig. S1 in Appendix A) is recommended. The Seq2Seq architecture consists of an encoder module and a decoder module. The encoder converts the loading path into dense tensors, which are referred to as contexts. The decoder autoregressively generates future responses using the historical responses and the derived contexts. The integration of the attention mechanism can significantly augment the ability of the constructed model to extract global historical information, thus mitigating the memory decay issue that is inherent in conventional recurrent neural networks. In addition, to fuse the information gleaned from the HGNN, the structural embeddings are concatenated with excitations at every time step. The mathematical formulation of the dynamic feature processor is as follows:

$$\mathbf{c}_\tau = \mathcal{F}_{\text{enc}}(\mathcal{A}(\mathbf{s}_{i \leq \tau}, \mathbf{s}_{i \leq \tau}) \cdot \mathcal{V}_{\text{enc}}(\mathbf{s}_{i \leq \tau})), \tau = 1, 2, \dots, T \quad (8)$$

$$\mathbf{u}_\tau^{\text{self}} = \mathcal{F}_{\text{dec}}^{\text{self}}(\mathcal{A}(\mathbf{r}_{i \leq \tau-1}, \mathbf{r}_{i \leq \tau-1}) \cdot \mathcal{V}_{\text{dec}}(\mathbf{r}_{i \leq \tau-1})), \tau = 1, 2, \dots, T \quad (9)$$

$$\mathbf{u}_\tau^{\text{cross}} = \mathcal{F}_{\text{dec}}^{\text{cross}}(\mathcal{A}(\mathbf{r}_{i \leq \tau-1}, \mathbf{s}_{i \leq \tau}) \cdot \mathcal{V}_{\text{dec}}(\mathbf{r}_{i \leq \tau-1})), \tau = 1, 2, \dots, T \quad (10)$$

$$\mathbf{r}_\tau = \mathcal{F}_{\text{out}}(\mathbf{u}_\tau^{\text{self}} \parallel \mathbf{u}_\tau^{\text{cross}}), \tau = 1, 2, \dots, T \quad (11)$$

where \mathbf{s} , \mathbf{r} , and \mathbf{u} represent the stimulus sequence, the structural response sequence, and the processed sequence in the middle layer, respectively; τ denotes the current time step during the prediction process (T steps in total); $\mathbf{s}_{i \leq \tau}$ denotes the stimulus information

from start to current τ ; \mathbf{r}_τ represents the structural responses at time step τ and $\mathbf{r}_{i \leq \tau-1}$ denotes the structural responses from start to the last time step $\tau-1$; \mathbf{c}_τ represents the contexts acquired from the encoder at time step τ ; $\mathbf{u}_\tau^{\text{self}}$ and $\mathbf{u}_\tau^{\text{cross}}$ represent the processed information at time step τ from self-/cross-attention layer, respectively. Furthermore, \mathcal{A} signifies the attention operator that utilizes a query tensor and a key tensor; \mathcal{F}_{enc} , $\mathcal{F}_{\text{dec}}^{\text{self}}$, and $\mathcal{F}_{\text{dec}}^{\text{cross}}$ signify feed-forward neural networks (FNN) of the encoder/decoder in self-/cross-attention mechanisms, respectively; the value functions \mathcal{V}_{enc} and \mathcal{V}_{dec} are the multilayer perceptrons (MLPs) that transform inputs into a form with proper dimensions in the encoder and decoder, respectively; \mathcal{F}_{out} is the output layer, and $(\cdot \parallel \cdot)$ denotes the tensor concatenation operation.

The fusion of the HGNN embedding model with the response generation module forms an end-to-end data-driven framework for predicting the full-range responses of structural systems. This framework operates on HetG representations, effectively digitizing diverse structural systems and comprehensively accommodating both intrinsic structural configurations and random external excitations.

4. Implementation

Following the proposed pipeline, we implement a computational model called StructGNN-N. An illustration of the model is shown in Fig. 3. More details are explained below.

4.1. HGIN: Heterogeneous graph isomorphism network

In a structural system, many joints share identical local topological structures but exhibit different mechanical behaviors during the analysis process. To capture the topological information contained in a graph and differentiate between various node features [20], a graph isomorphism network (GIN) is established [53]. This network uses MLPs to approximate the functions \mathcal{M} , \mathcal{G} , and \mathcal{U} based on the theory of the graph isomorphism problem, ensuring the injectiveness of feature mapping during the message passing process. Furthermore, we introduce the joint representation learning paradigm [54] to combine heterogeneous node types as well as features to form a common latent space, as shown in Fig. 3(a). Specifically, additional MLP layers with sum pooling are employed to accommodate the various dimensions of nodes, enabling the integration of heterogeneous information derived from neighborhoods. Therefore, we construct a modified GIN model named a heterogeneous GIN (HGIN):

$$\mathbf{h}_i^{(l)} = \text{MLP}_{\mathbf{v}_i}^{(l)}((1 + \epsilon^{(l)}) \cdot \text{MLP}_{\mathbf{v}_i}^{(0)} \mathbf{v}_i + \sum_{a_{ij}=1} \text{MLP}_{\mathbf{v}_j}^{(0)} \mathbf{v}_j) \quad (12)$$

$$\mathbf{h}_i^{(l)} = \text{MLP}_{\mathbf{v}_i}^{(l)}((1 + \epsilon^{(l)}) \cdot \mathbf{h}_i^{(l-1)} + \sum_{a_{ij}=1} \mathbf{h}_j^{(l-1)}), l > 1 \quad (13)$$

where $\epsilon^{(l)}$ is a learnable parameter for adjusting the weight of the node feature \mathbf{v}_i (if $\epsilon^{(l)} = 0$, the weight of the information derived from \mathbf{v}_i is the same as that of its neighbor node); $\text{MLP}_{\mathbf{v}_i}^{(l)}$ and $\text{MLP}_{\mathbf{v}_j}^{(l)}$ are the MLP layers for the corresponding node i and node j in the l th layer. For different types of nodes, different MLP layers are trained to aggregate information and enable the unification of the various dimensions of the initial node embeddings in the hidden layers.

4.2. Mechformer: A transformer for mechanical computations

For the dynamic feature module, we introduce the mechanical transformer (Mechformer) model (Fig. 3(b)) proposed by the

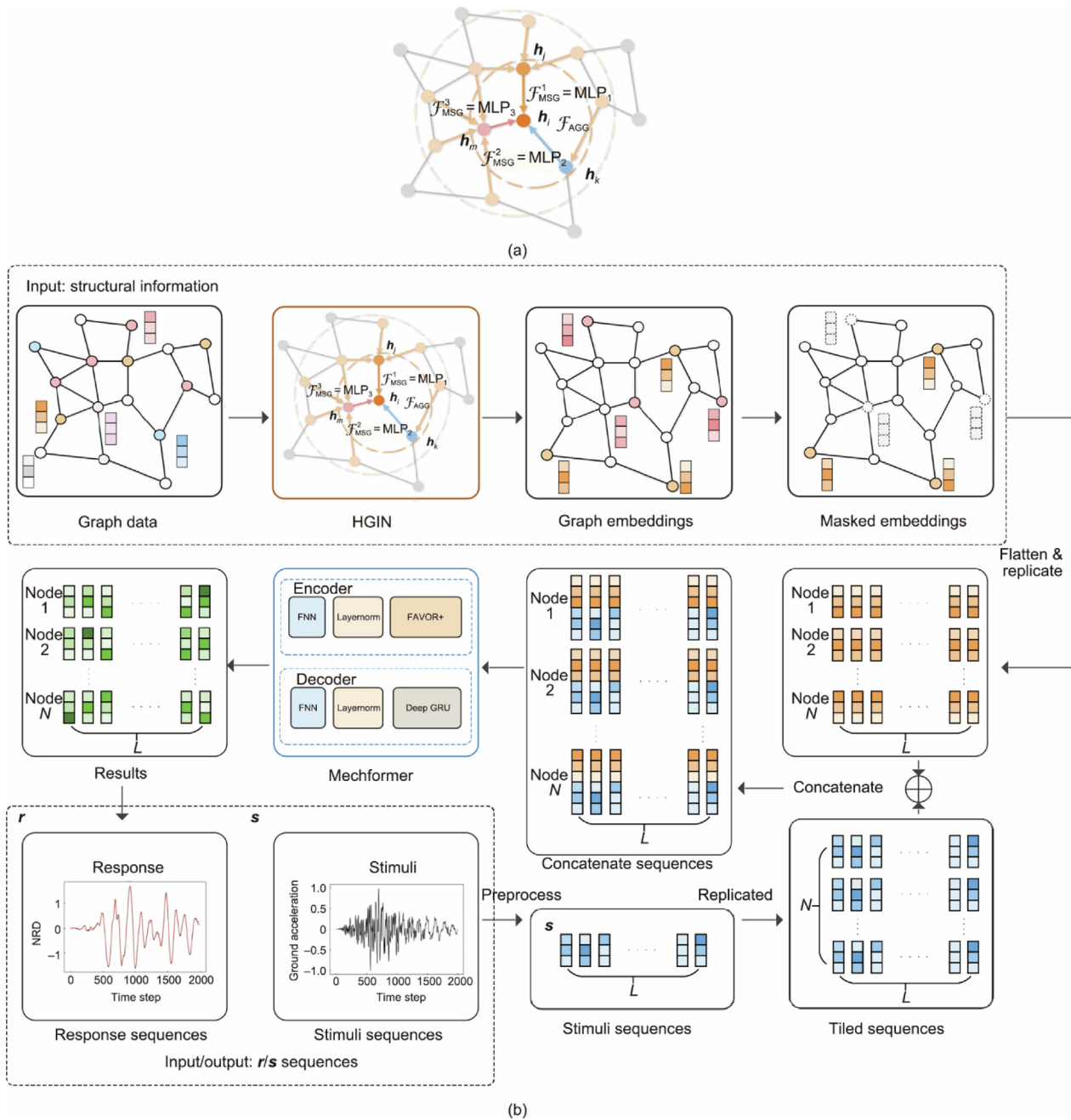


Fig. 3. Illustrations of the HGIN module and StructGNN-N model. (a) Joint representation learning in the HGIN. The state information h_i is computed from of neighborhood state information h_j , h_m , and h_k ; \mathcal{F}_{MSG}^1 , \mathcal{F}_{MSG}^2 , and \mathcal{F}_{MSG}^3 are the messaging (MSG) FNNs using different MLP layers (MLP₁, MLP₂, and MLP₃); \mathcal{F}_{AGG} is the aggregating (AGG) FNN. (b) The architecture of the StructGNN-N. L is the sequence length; GRU refers to the gated recurrent unit and FAVOR+ is the positive orthogonal random feature algorithm.

authors in Ref. [46] (Fig. S2 in Appendix A). The Mechformer was derived from the transformer architecture [52] and belongs to the Seq2Seq model family.

To extract the essential information from a sequence, a multi-head attention mechanism can discover the correlation between any two positions within the sequence. In the vanilla transformer [32], the attention results obtained for the query (\mathbf{Q}), key (\mathbf{K}), and value (\mathbf{V}) tensors are calculated as follows:

$$A(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \cdot \mathbf{V}, O(L^2d) \quad (14)$$

However, the temporal and spatial costs $O(L^2d)$ are proportional to the square of the sequence length L and the model dimension d , which imposes a high memory demand, especially with the $\langle s, r \rangle$ sequences in nonlinear analyses involving large L values. To achieve improved efficiency, the fast attention via FAVOR+ algorithm [55] is employed in the Mechformer model to replace the standard attention module (Fig. S3 in Appendix A). By approximating the nonlinear softmax activation function in Eq. (14) with a linear combination calculated by the kernel function, the complexity level is reduced to $O(Lmd)$:

$$\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \approx \mathbf{Q}'\mathbf{K}'^T = \phi(\mathbf{Q})\phi(\mathbf{K})^T \quad (15)$$

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \approx \mathbf{Q}'\mathbf{K}'^T\mathbf{V}, O(Lmd) \quad (16)$$

where $\phi: \mathbb{R}^d \rightarrow \mathbb{R}_+^m$ is a random feature mapping kernel function; m denotes the hidden dimensionality and $m \ll L$. \mathbf{Q}' and \mathbf{K}' are the mapped query and key in the hidden dimension. Furthermore, considering the causality observed in structural analyses, the future information contained in the sequences is masked using the prefix summation algorithm, which corresponds to computing the lower-triangular part of the attention matrix.

To integrate the structural information \mathbf{h}_i derived from the HGIN with the j th input stimulus sequence \mathbf{s}^j of the Mechformer, the structural embeddings of node i are first replicated to match the sequence length L of \mathbf{s}^j , obtaining replicated embeddings \mathbf{H}_i :

$$\mathbf{H}_i = \underbrace{(\mathbf{h}_i, \mathbf{h}_i, \dots, \mathbf{h}_i)}_L \quad (17)$$

Then, they are concatenated with \mathbf{s}^j :

$$\mathbf{H}_i^j = (\mathbf{s}^j, \|\mathbf{h}_i\|) = \begin{bmatrix} \mathbf{s}_1^j & \mathbf{s}_2^j & \dots & \mathbf{s}_L^j \\ \mathbf{h}_i & \mathbf{h}_i & \dots & \mathbf{h}_i \end{bmatrix} \quad (18)$$

where \mathbf{H}_i^j denotes the concatenated information of node i and the stimulus sequence \mathbf{s}^j ; $\mathbf{s}_1^j, \mathbf{s}_2^j, \dots, \mathbf{s}_L^j$ represent the information of \mathbf{s}^j corresponding to the step $1, 2, \dots, L$.

4.3. Masked response training strategy

We employ a data-driven inference paradigm, training the model based on the nonlinear response sequences of structural systems subjected to different input stimulus sequences. Specifically, for a given structural system with N nodes, the structural responses are denoted as $\{\mathbf{r}_1^j, \dots, \mathbf{r}_N^j\}$, where \mathbf{r}_i^j is the structural response sequence corresponding to node i produced for the input stimuli \mathbf{s}^j . The dataset for this structural system can be represented as shown below:

$$\{\mathbf{H}, \mathbf{s}^j, \mathbf{r}_1^j, \dots, \mathbf{r}_N^j | j \in \{1, 2, \dots, K\}\} \quad (19)$$

where K denotes the total number of stimulus sequences (e.g., various seismic waves). In practice, experimental data for structural systems are extremely scarce, whereas generating data using an FE analysis is a computationally intensive procedure. Consequently, fully exploiting the information derived from existing data is highly important. To augment the data utilization process, we propose a masked response training strategy.

The data of a structural system is split into N groups, each in the form of <structural node embedding, node stimulus sequence, node response sequence>:

$$\{\mathbf{h}_i, \mathbf{s}^j, \mathbf{r}_i^j | i \in \{1, 2, \dots, N\}, j \in \{1, 2, \dots, K\}\} \quad (20)$$

A subset of the nodes (size = n) is selected for the training set (TR):

$$\text{TR} = \{\mathbf{h}_i, \mathbf{s}^j, \mathbf{r}_i^j | i \in \text{TR}_n, j \in \{1, 2, \dots, K\}\} \quad (21)$$

$$\text{TR}_n = \{i_1, i_2, \dots, i_n\} \subset \{1, 2, \dots, N\}, \quad n < N \quad (22)$$

where TR_n denotes the subset of the selected nodes.

The remaining nodes are used as the validation/test set (TE):

$$\text{TE} = \{\mathbf{h}_i, \mathbf{s}^j, \mathbf{r}_i^j | i \in \{1, 2, \dots, N\} \setminus \text{TR}_n, j \in \{1, 2, \dots, K\}\} \quad (23)$$

Typically, the number of nodes N is significantly larger than the number of structural system samples. Inspired by this observation,

we break the structural-level data down into node-level data to facilitate the comprehensive extraction of the implicit mechanisms within the data, thereby significantly enlarging the trainable data size and alleviating the data scarcity burden. Additionally, by partitioning partial nodes (and their connectivity) as the training set, our strategy enables the prediction of the whole system from local inputs. This provides a new solution for smart virtual sensing [56].

Programmatically, we design a masking technique to unify the computation process. The mask \mathbf{M}^{TR} is a matrix with dimensions of $n \times N$ that performs multiplication for random sampling tasks:

$$\begin{bmatrix} \mathbf{r}_{i_1}^j \\ \mathbf{r}_{i_2}^j \\ \vdots \\ \mathbf{r}_{i_n}^j \end{bmatrix} = \mathbf{M}^{\text{TR}} \mathbf{r}^j = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & \dots & m_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \dots & m_{NN} \end{bmatrix} \begin{bmatrix} \mathbf{r}_1^j \\ \mathbf{r}_2^j \\ \vdots \\ \mathbf{r}_N^j \end{bmatrix}, \quad m_{ij} = \begin{cases} 1, & j = i_n \\ 0, & j \neq i_n \end{cases} \quad (24)$$

where i_1, i_2, \dots, i_n are the selected indexes, which correspond to the selected nodes i_1, i_2, \dots, i_n ; $\mathbf{r}_{i_n}^j$ is the vector of \mathbf{r}^j (the j th structural response sequence) corresponding to index i_n , m_{ij} is an element in \mathbf{M}^{TR} ($i = 1, \dots, N$; $j = 1, \dots, N$), and each row of \mathbf{M}^{TR} is a unit vector. By modifying the mask matrix, we can determine the size and distribution of the training set. The mask matrix \mathbf{M}^{TR} is also applied to the structural embedding \mathbf{H}^j :

$$\begin{bmatrix} \mathbf{H}_{i_1}^j \\ \mathbf{H}_{i_2}^j \\ \vdots \\ \mathbf{H}_{i_n}^j \end{bmatrix} = \mathbf{M}^{\text{TR}} \mathbf{H}^j = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & \dots & m_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \dots & m_{NN} \end{bmatrix} \begin{bmatrix} \mathbf{H}_1^j \\ \mathbf{H}_2^j \\ \vdots \\ \mathbf{H}_N^j \end{bmatrix}, \quad m_{ij} = \begin{cases} 1, & j = i_n \\ 0, & j \neq i_n \end{cases} \quad (25)$$

where \mathbf{H}_i^j represents the concatenated information of node i ($i = 1, \dots, N$) and the stimulus sequence \mathbf{s}^j , and $\mathbf{H}_{i_1}^j, \mathbf{H}_{i_2}^j, \dots, \mathbf{H}_{i_n}^j$ are the concatenated information of the selected nodes i_1, i_2, \dots, i_n , respectively.

5. Validation

5.1. Data preparation

5.1.1. Basic information of the structural system

For demonstration purposes, a real-world 9-floor office building is chosen. As shown in Fig. 4, this building is an L-shaped RC structure with an outline of 52.8 m (length) \times 43.2 m (width). The height of each floor ranges from 4.8 to 5.4 m.

The first three natural frequencies of the structure are 0.97, 1.55, and 2.19. The main structural components include RC columns, slabs, and beams. The sections of these components vary in their sizes and reinforcement ratios. The corresponding graph data representation contains 4303 nodes of four types and 6948 edges. Some of the node features of the joints and the beam components are listed in Tables 1 and 2.

5.1.2. Data preparation for stimulus and response sequences

We aim to train a model that can conduct a nonlinear analysis on a structural system subjected to seismic waves. The stimulus sequences are the ground acceleration records of different earthquakes, which act upon the structure in the horizontal direction. Many ground motion records are sampled from the Pacific Earthquake Engineering Research (PEER) database [57], and a total of 70 records that match the structural frequency characteristics of the target building are selected, covering a relatively wide range

of seismic parameters. A portion of the ground motion records is shown in Table 3.

A numerical model of the structure is established using the fiber beam model [58,59] to obtain $\langle s, r \rangle$ sequence pairs. The time inter-

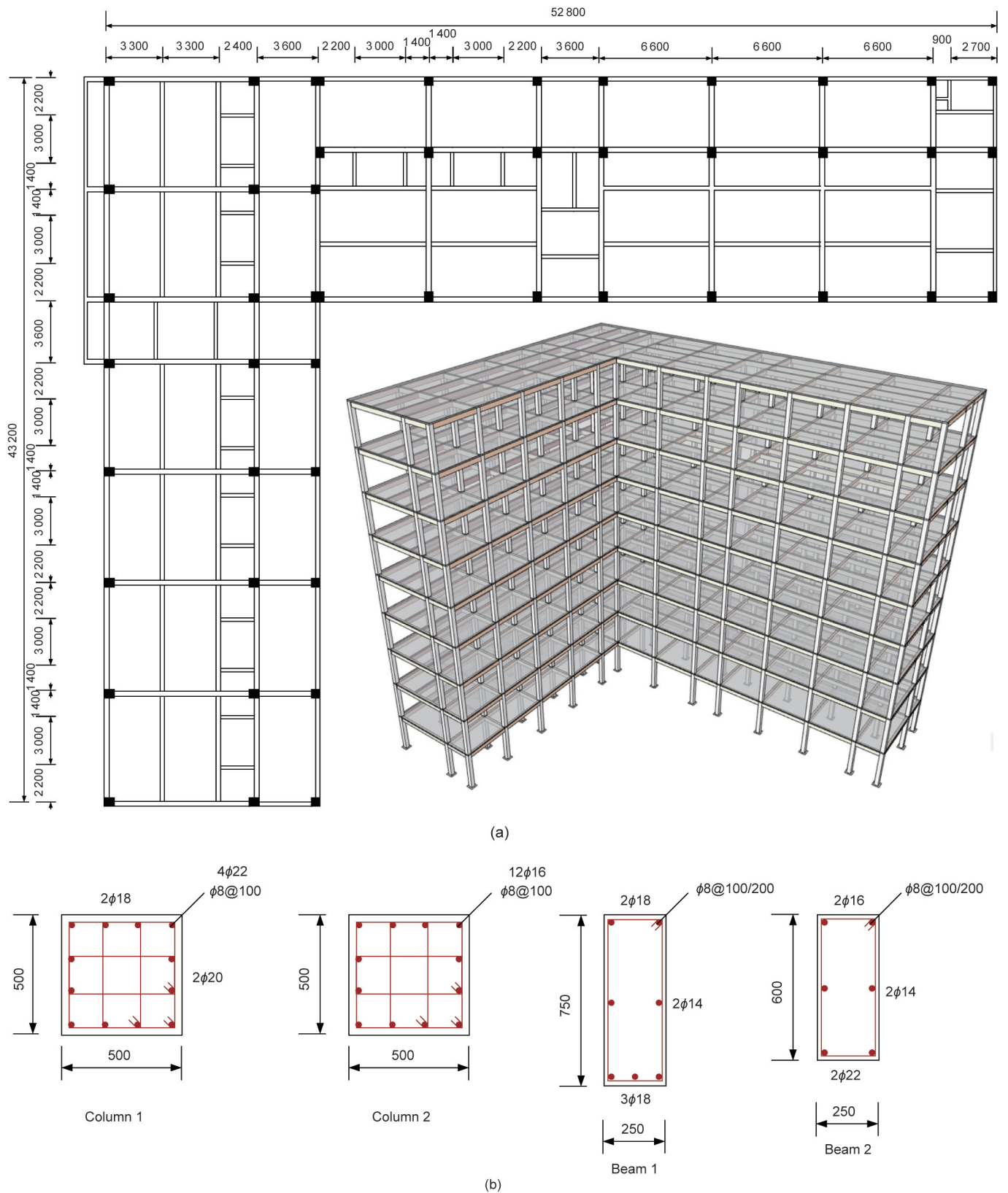


Fig. 4. Illustration of the example structural system (unit: mm). (a) Three-dimensional (3D) view and plan layout of the example structural system; (b) dimensions of the main structural component sections.

Table 1
Some of the node features of joints.

Feature	Unit	Detail
i	—	Identifier (ID) of a node
x	mm	X coordinate of the node position relative to node 0
y	mm	Y coordinate of the node position relative to node 0
z	mm	Z coordinate of the node position relative to node 0 (the height of a node relative to the ground)
$floor_i$	—	The floor in the system to which the node belongs

Table 2
Some of the node features of the beam components.

Feature	Unit	Detail
i, j	—	IDs of nodes connected by a beam
(dx, dy, dz)	—	Unit vectors denoting the directions of the different components in the space
L_{beam}	mm	The valid length of the beam
h	mm	The valid height of a section of the beam
b	mm	The valid width of a section of the beam
ρ	$g \cdot mm^{-3}$	The equivalent density of the beam when the load (including the slab on the beam) is transformed into the self-weight of the component. For a concrete component with no load, $\rho = 2.5$
r_x	—	The ratio of the area of steel reinforcement bars in the x direction to the total section area
r_y	—	The ratio of the area of steel reinforcement bars in the y direction to the total section area
f_c	MPa	The designed compressive strength of the concrete
f_t	MPa	The designed tensile strength of the concrete
ϵ_{cu}	—	The ultimate compressive strain of the concrete
E_c	MPa	The designed elasticity modulus for the concrete
f_y	MPa	The yield strength of the steel reinforcement bars
f_u	MPa	The ultimate strength of the steel reinforcement steel bars
E_s	MPa	The designed elasticity modulus of the steel reinforcement bars

val is set as 0.02 s, ensuring that the sequence lengths range from 1000 to 2500. The simulation of each ground motion record in the FE analysis takes approximately 1 to 3 h. The time series of the displacement of each node is extracted to form a dataset. Fifty pairs of $\langle \mathbf{s}, \mathbf{r} \rangle$ sequences are randomly selected for training, 10 pairs are used for validation, and the rest are employed for testing. The computing length of the model is set to 2048. Therefore, sequences with shorter lengths are padded with the end values at their tails, while the longer sequences are truncated. The data generation process runs on 10 computers in parallel and requires approximately 64 h of computations.

5.2. Model configuration

A three-layer HGIN is employed for StructGNN-N. The hidden dimensions of the HGIN and the Mechformer module are 32. The

Table 3
Parameters of selected ground motion records from the PEER database.

Earthquake	Year	Monitoring station	Richter scale	V_{s30} ($m \cdot s^{-1}$)	PGA (g)	MSE
“Imperial Valley-02”	1940	“El Centro Array #9”	6.95	213.44	0.281	0.020
“Imperial Valley-06”	1979	“El Centro Array #8”	6.53	206.08	0.610	0.028
“Northridge-01”	1994	“Pacific Palisades-Sunset”	6.69	191.06	0.460	0.050
“Northern Calif-03”	1954	“Ferdale City Hall”	6.50	219.31	0.163	0.050
“Loma Prieta”	1989	“Gilroy Array #4”	6.93	221.78	0.419	0.041

V_{s30} : average shear wave velocity in the top 30 m; PGA: peak ground acceleration; MSE: mean squared error.

dimension of the kernel function employed in FAVOR+ is 64. The encoder and decoder modules in the Mechformer have three blocks. A two-head attention mechanism is employed, with a hidden size of 128 in the linear layer. Layer normalization [60] is added. In total, the model has 102 093 parameters.

For optimization purposes, we use the adaptive moment estimation (Adam) [61] optimizer with an initial learning rate of 0.005. The model is trained for 1000 epochs, and the learning rate decays 50% after every 20% of the training epochs. The mean squared error (MSE) loss is employed to measure the training results:

$$Loss = \text{mean}(\|\mathcal{R}_{pred} - \mathcal{R}_{label}\|_2) \tag{26}$$

where \mathcal{R}_{pred} and \mathcal{R}_{label} represent the results predicted by the model and the labeled results used for reference, respectively, and $\|\cdot\|_2$ is the L2 norm. For validation and testing, the loss is transformed into a relative accuracy term (Accu):

$$Accu = 1 - \left(\frac{\|\mathcal{R}_{pred} - \mathcal{R}_{label}\|_2}{\|\mathcal{R}_{label}\|_2 + \eta} \right)^{1/2} \tag{27}$$

where η is a small value that is used to avoid division by zero.

The initial graph data are directly fed into the HGIN module. After concatenating them with the training stimulus sequences, the combined features of 256 nodes are randomly chosen as a batch using a random mask matrix \mathbf{M}^{TR} , thus modeling the shuffling process of minibatch training. The model is trained on the PyTorch [62,63] platform using an American NVIDIA RTX 4080 graphics processing unit (GPU) with 16 GB of memory. The training and validation processes employed for the demonstrated structural system require approximately 3 h.

5.3. Model performance

The trained model achieves a relative accuracy level of 97.5% on the test dataset. The predictions produced under different seismic waves are compared with the results of the FE analysis, as illustrated in Fig. 5, where \mathcal{R} refers to the result displacement, NRD refers to the normalized displacement \mathcal{R}_{NRD} relative to the ground displacement \mathcal{R}_{GD} :

$$\mathcal{R}_{NRD} = \frac{\mathcal{R} - \mathcal{R}_{GD}}{\max(\text{abs}(\mathcal{R}_{GD}))} \tag{28}$$

NGD refers to the normalized ground displacement \mathcal{R}_{NGD} :

$$\mathcal{R}_{NGD} = \frac{\mathcal{R}_{GD}}{\max(\text{abs}(\mathcal{R}_{GD}))} \tag{29}$$

StructGNN-N is able to predict the full-range displacement sequences with considerable accuracy and successfully distinguishes between different positions rather than producing isolated results. The model exhibits excellent generalizability across

various seismic waves, enabling the identification of potential weak points using comprehensive response information rather than relying on several static mechanical indices. Additionally,

the relative lateral displacement and interstory drift values at the peak points are extracted and compared with the results acquired from the FE models. As shown in Fig. 6, the results computed by

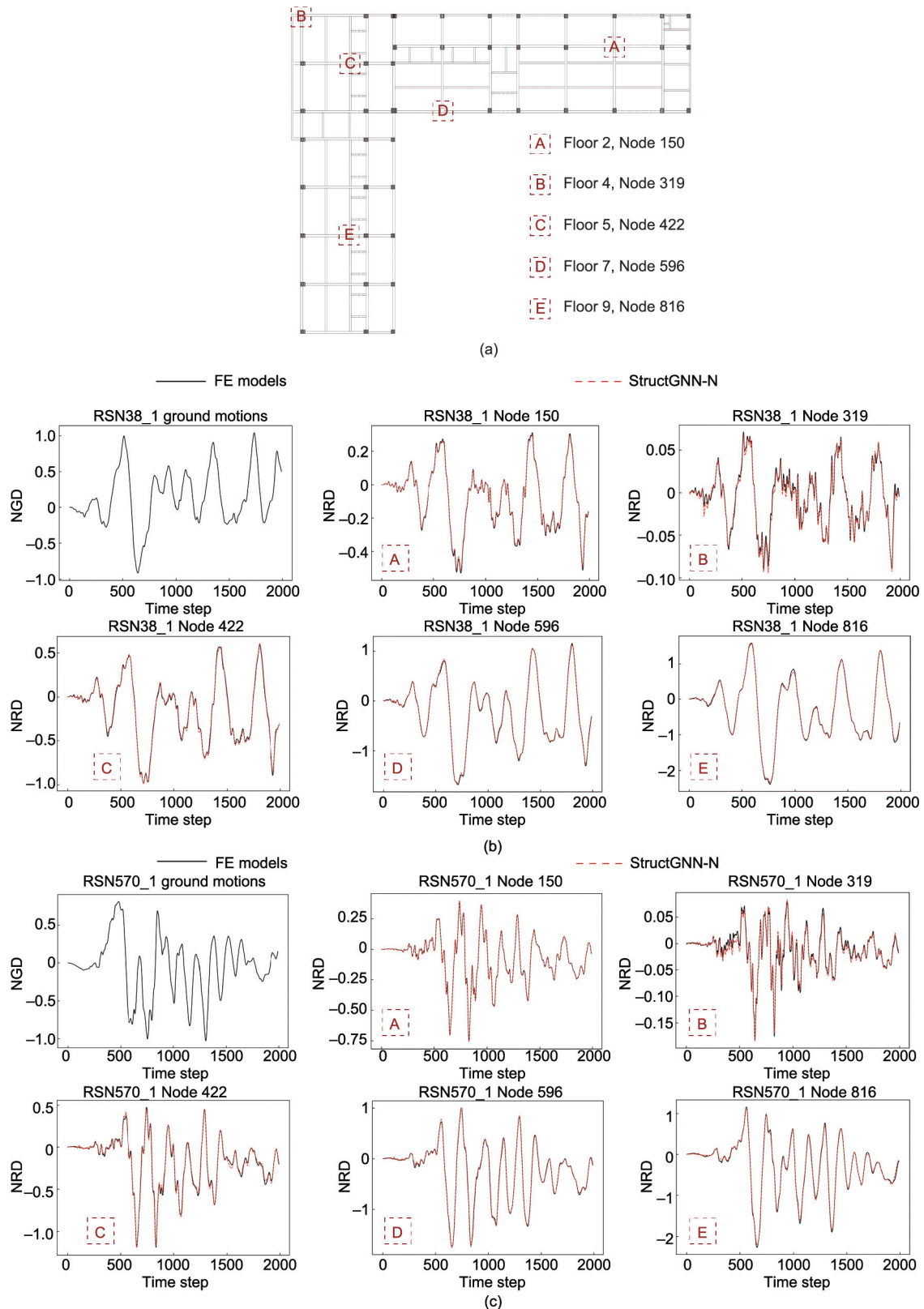


Fig. 5. Comparison between the full-range results of the StructGNN-N and FE models. (a) Distribution of the selected positions in the structural system; (b) input stimulus sequence: RSN38_1; (c) input stimulus sequence: RSN570_1.

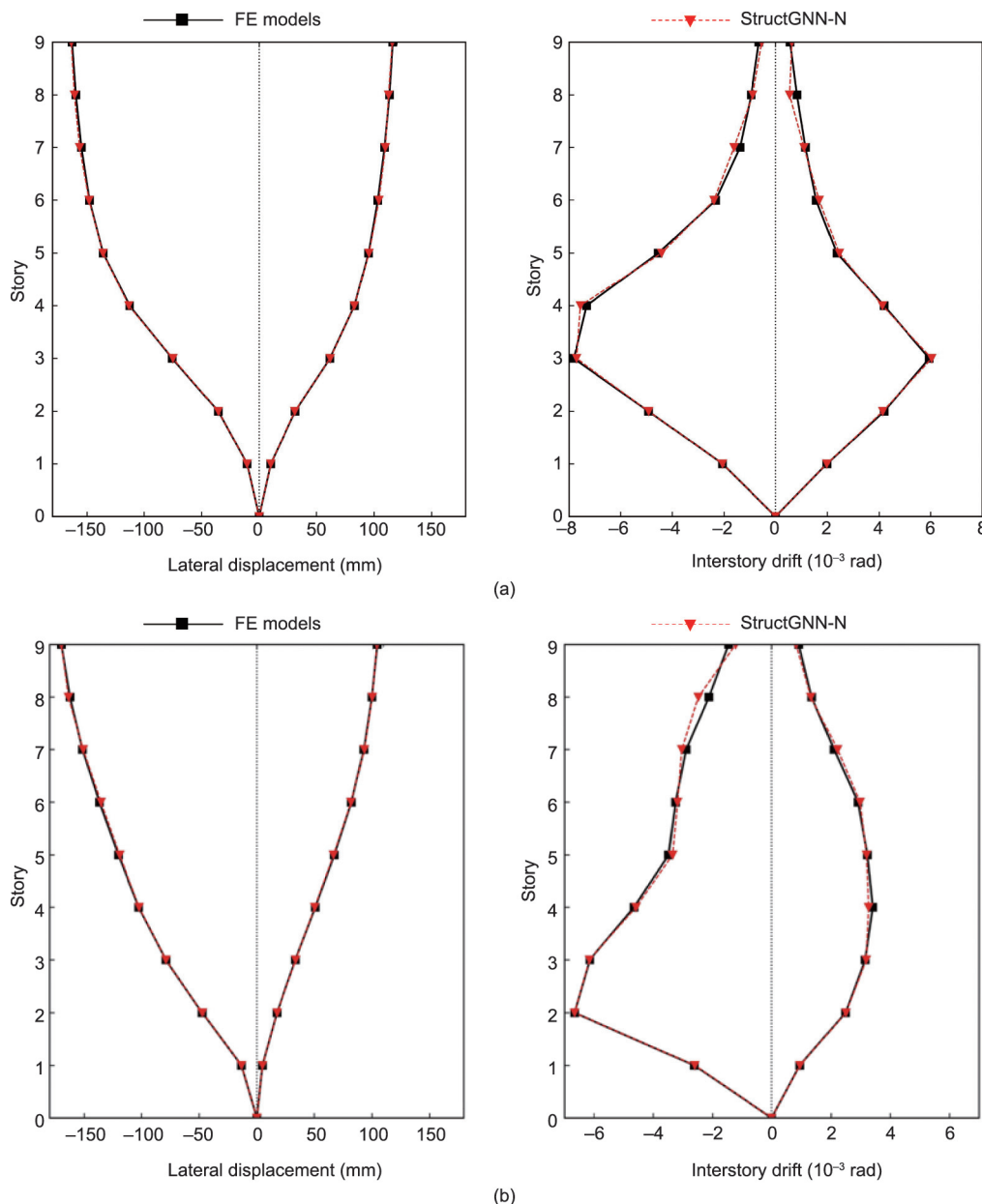


Fig. 6. Comparison between the lateral displacement and story drift values of the StructGNN-N and FE models. (a) Input stimulus sequence: RSN38_1; (b) input stimulus sequence: RSN570_1.

StructGNN-N closely align with those of the FE analysis. The Mechformer module captures the cyclic loading characteristics in the seismic analysis task, accurately delineating different loading and unloading phases. While minor discrepancies appear during the initial and final stages of the seismic time history analysis, within the range of the smaller structural response amplitudes, at critical peak points that potentially lead to structural failures, such as the maximum interstory drift, the model predicts responses with greater accuracy. This is attributed to the attention mechanism, which, through extensive data training, focuses more on the historical influences of the peak values.

Overall, the numerical experiment validates the effectiveness of the StructGNN-N model in terms of reproducing history-dependent hysteresis loops as well as its excellent ability to capture structural characteristics. Notably, the trained model takes several seconds to predict responses for the whole structure, sig-

nificantly improving upon the temporal efficiency of the traditional FE methods.

5.4. Structural generalization

For generalization purposes, the StructGNN-N model is trained on a dataset including 942 real-world engineering projects to predict seismic responses across different structures subjected to various ground accelerations. Owing to computational resource and time limitations, 10 ground motion records are selected from the PEER dataset for each structural system. The lateral displacements and interstory drifts at the point with the maximum structural response are extracted and compared with the FE analysis results, achieving an average accuracy of 95.4%. The representative test cases are detailed below.

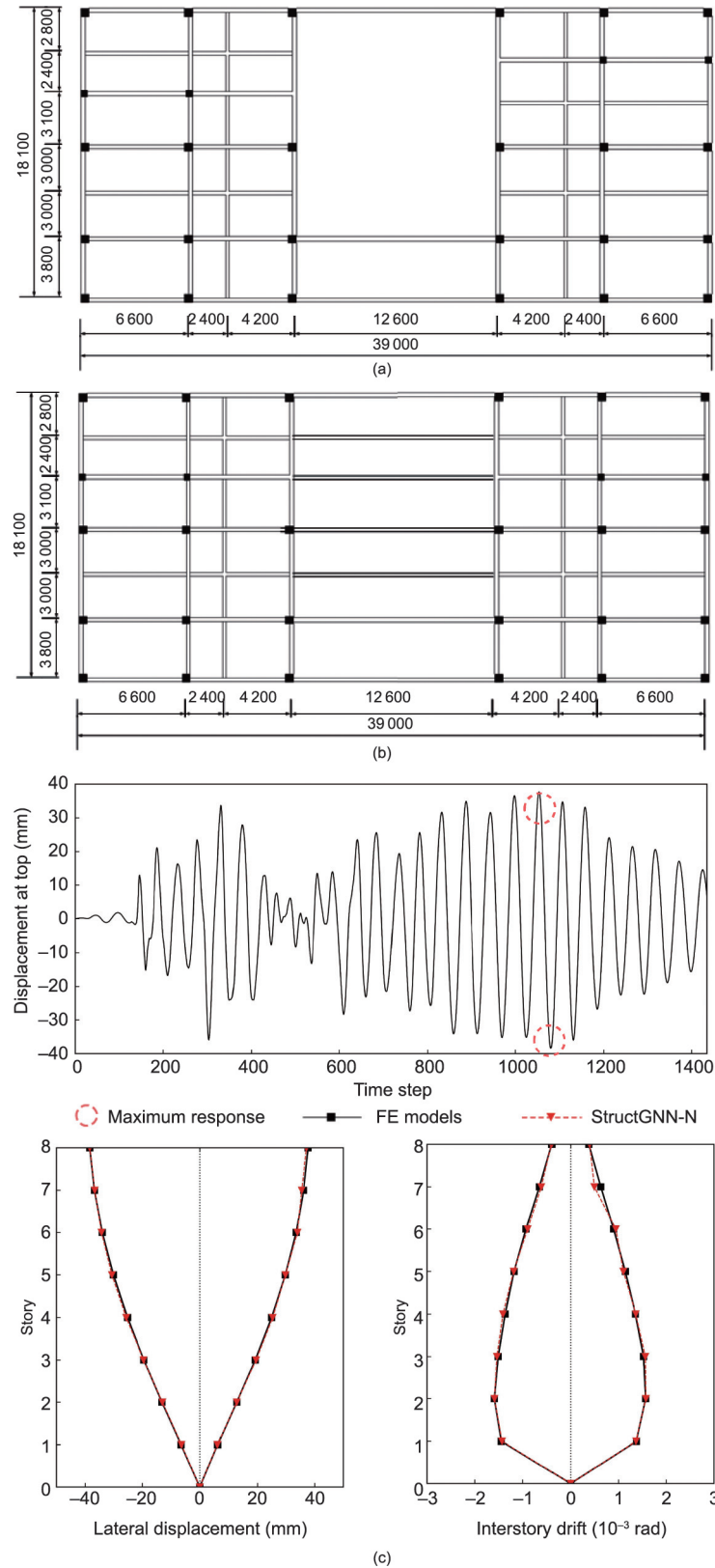


Fig. 7. Analysis results obtained for Case 1. (a) Plan view of Case 1: 1st–2nd floors (unit: mm); (b) plan view of Case 1: 3rd–8th floors (unit: mm); (c) input stimulus sequence: RSN167_1.

(1) **Case 1: a structure with a regular plan layout.** Case 1 involves an eight-story RC structure with a regular and symmetrical plan layout, as depicted in Fig. 7. A full-range nonlin-

ear analysis is conducted under seismic loading conditions, yielding accurate lateral displacement and interstory drift predictions.

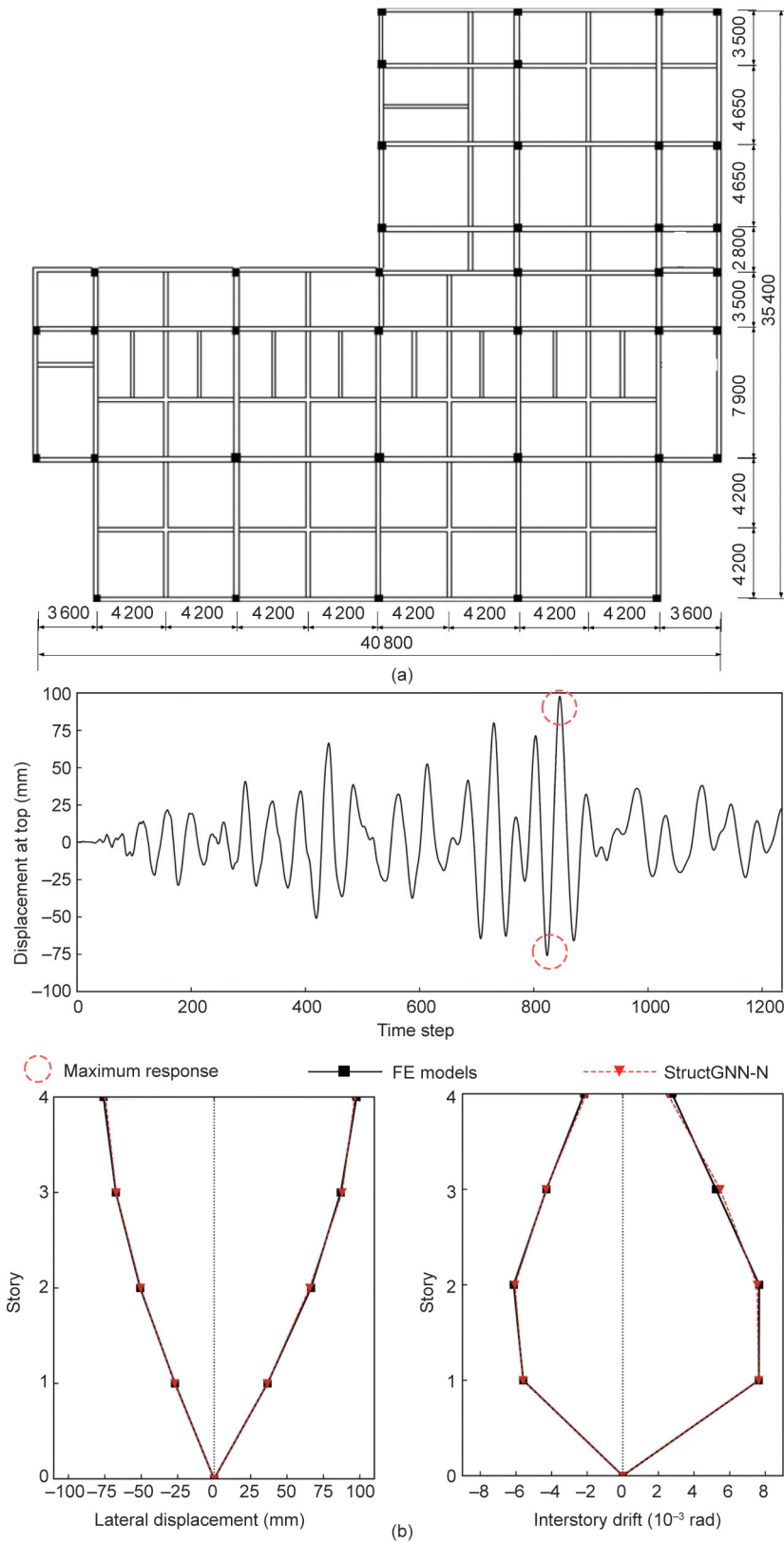


Fig. 8. Analysis results obtained for Case 2. (a) Plan view of Case 2 (unit: mm); (b) input stimulus sequence: RSN9_2.

(2) **Case 2: a structure with an irregular plan layout.** Case 2 features a four-story RC structure characterized by an asymmetrical and irregular layout, thus diverging from the training dataset.

The computational results shown in Fig. 8 accurately capture the asymmetry and demonstrate the maximum lateral displacement and story drift of the structure.

6. Discussions

6.1. Parameter analysis

The StructGNN-N model can infer information about an entire structural system on the basis of data acquired from selected points within the structure. A critical issue is determining how many selected points are adequate for predicting the responses of the entire structure. Answering this question also inspires us to strategically place measurement devices at the optimal positions to obtain the mechanical responses at these locations and employ StructGNN-N to assess its overall structural performance. In practical engineering applications, economic considerations and technical limitations often restrict the numbers and locations of sensors. StructGNN-N can be trained on a subset of data derived from these limited points and then applied to predict responses at other locations where direct measurements are not available, aligning with the concept of virtual sensing [56].

To address this issue, we investigate the effects of the selection and distribution of the training dataset consisting of n nodes on the resulting model performance. Various percentages of training data (unmasking ratio: n/N) and different selection approaches, which

correspond to different settings for the measurement instruments, are considered. ① RD: The training samples are randomly distributed in the graph, corresponding to randomly placed measurement instruments within the structural system. ② UD: The training samples are uniformly distributed in the graph, corresponding to placing the measurement instruments within the structural system by following a specific pattern. ③ CL: The samples are concentrated in the lower part of the structural system. ④ CU: The samples are concentrated in the upper part of the structural system. In accordance with the corresponding selection approaches, different masks are applied to the node embeddings, as shown in Fig. 9(a).

The experiment is carried out on models with different configurations: S32 is the StructGNN-N model described in the previous section, and S16 and S64 are lighter and heavier versions, respectively, with different hidden dimensions. The total numbers of parameters required for these models are 26 493, 102 093, and 400 749. To ensure a fair comparison, all the models are fully trained for 1000 epochs and tested on the same dataset while using Eq. (27) as the metric. The results are presented in Fig. 9. The RD and UD strategies yield similar results with the same model configuration, which aligns with our intuition when the structural

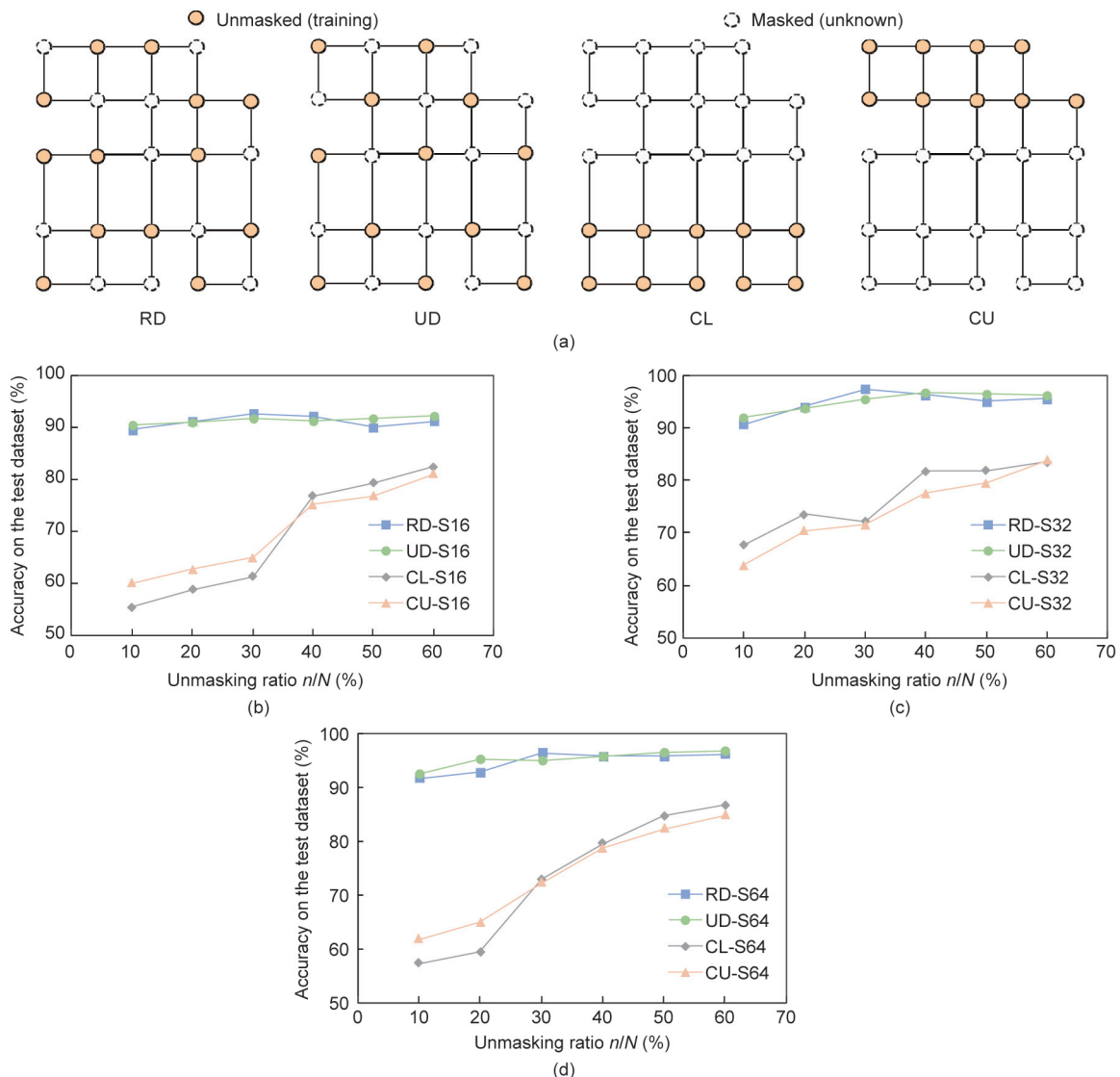


Fig. 9. Comparison among the prediction accuracies of models S16, S32, and S64 with different masking strategies. (a) Different masks applied to the node embeddings; (b) Model S16; (c) Model S32; (d) Model S64.

system is sufficiently large. The key difference between the two strategies is that the RD strategy results in greater variability, whereas the UD strategy generally results in an upward trend in performance as the unmasking ratio increases. On the other hand, the CL and CU strategies yield less accurate results in general. The performance of these methods significantly improves as the unmasking ratio increases, but increasing the proportion of selected positions to over 50% is often impractical in real-world applications. Notably, even with only a 10%–20% unmasking ratio (which is often the case in engineering applications), the RD and UD strategies can achieve approximately 90% accuracy, which is

sufficient for obtaining a reasonable assessment of a structural system.

6.2. Ablation study

StructGNN-N is composed of two main parts: an HGIN for encoding structural information and the Mechformer module for sequence processing. In this section, we carry out an ablation study on these two components.

To encode the structural information, we adopt a DNN, a 2D CNN [64], and a vanilla gated recurrent unit (GRU) [65] for comparison purposes. The node features contained in the graph data of the examined structural systems are modified to adapt to the input (Fig. S4 in Appendix A). For sequence processing, we compare the Mechformer with the GRU and the vanilla transformer [52]. The unmasking ratio is 20%, and the same RD strategy is applied. All the models are trained for 1000 epochs and tested on the same dataset. The results are shown in Table 4.

Table 4 shows that StructGNN-N (HGIN + Mechformer) yields the optimal results, and the Mechformer achieves the highest accuracy in most combinations with different encoding modules. The results reveal that the encoding process of structural information is crucial for predicting the nonlinear responses of structural systems. The DNN encodes the features of each component individually and completely neglects their relationships, which limits its ability to generalize to the test dataset. The CNN encodes the features of each component and its “adjacent” neighbors in a grid setting [66].

Table 4
Comparison among the prediction accuracies achieved with different model configurations.

Module combination	Accuracy (%)
DNN + GRU	81.4
DNN + Transformer	87.9
DNN + Mechformer	88.3
CNN + GRU	72.5
CNN + Transformer	79.4
CNN + Mechformer	78.7
GRU + GRU	75.1
GRU + Transformer	83.4
GRU + Mechformer	84.2
HGIN + GRU	83.6
HGIN + Transformer	92.3
HGIN + Mechformer	94.2

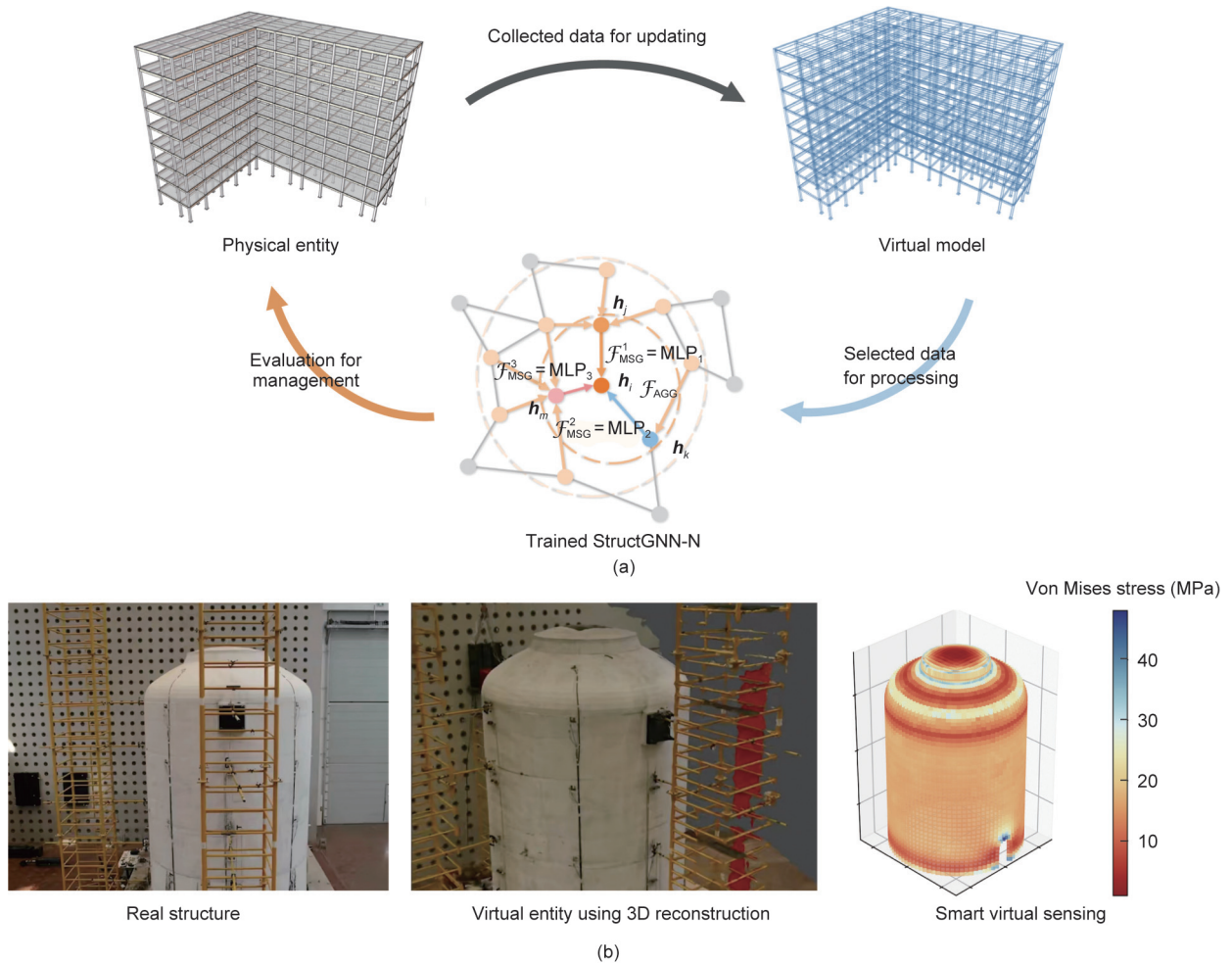


Fig. 10. Illustration of the StructGNN-N application related to smart virtual sensing in digital twins. (a) Application procedure of StructGNN-N for a digital twin; (b) case study concerning the structure of a nuclear containment shell.

However, real-world structural systems are typically too complex to be arranged in a grid format, and the “adjacent” neighbors in the grid may not be adjacent to the actual structure of interest. Additionally, determining the optimal kernel size and stride of the convolution is challenging. The GRU module processes the features of each component in sequential order, resulting in variable outcomes each time and lower efficiency than that of the other modules. In contrast, the HGIN module fully utilizes the physical connections among the different components of structural systems, and the Mechformer is better adapted to mechanical computations than the vanilla transformer is, thereby attaining the highest accuracy.

6.3. Potential applications

We have demonstrated the effectiveness of StructGNN-N for conducting nonlinear analyses on structural systems. The training and evaluation processes of StructGNN-N align with the task of building digital twins of structural systems in engineering applications [67].

To construct a digital twin, information about different components within the target structural system is collected throughout the entire life cycle of the structure via the virtual sensing technique. This physical information, including mechanical responses induced during the service life, is critically associated with safety assessments [68,69]. As shown in Fig. 10, a virtual model is updated to reflect the corresponding real entity on the basis of the collected information, and adjustments are made to manage the physical entity according to the discoveries made by the virtual model. This process dynamically maintains the circular flow of data between the virtual model and the physical entity. Owing to its powerful simulation capabilities, the fully trained StructGNN-N approach can be integrated into the digital twin paradigm with conventional simulation platforms to process real-world mechanical data and quickly obtain nonlinear analysis results. Furthermore, even when limited data is acquired from the target structural system, which is often the case since collecting comprehensive information is a time-consuming and costly process during the service period, the fully trained StructGNN-N model can provide a timely and reasonable assessment of the entire structure for decision-making process.

7. Conclusions

In this paper, a novel method based on HetG DL is proposed to address the task of efficiently conducting nonlinear analyses on structural systems. The model trained based on the proposed method can be applied to different structures with diverse designs and predict the full-range history-dependent responses of these structure. The main contributions of this study are summarized as follows.

(1) We design a universal graph representation scheme for structural systems. HetG data comprehensively integrate information about different components within the target structural system and characterize the connectivity relationships among them.

(2) We propose a composite feature learning method to realize end-to-end nonlinear analyses of structural systems. An HGNN is introduced for encoding the internal structural information, and the Seq2Seq framework is recommended for predicting history-dependent full-range nonlinear responses.

(3) We implement HGIN and Mechformer models based on the proposed method, forming a structural computational model named StructGNN-N. This model distinguishes between nodes with similar local topologies and extracts the global historical dependencies underlying $\langle s, r \rangle$ sequences with linear complexity.

(4) We design a masked response-based training strategy for StructGNN-N. By masking the information of part of the target structural system, this from-part-to-whole paradigm enables sig-

nificant data augmentations, greatly mitigating the data scarcity issue at the structural system level.

(5) Numerical experiments validate the effectiveness of the proposed method. The StructGNN-N model successfully generalizes to different seismic stimuli and structures, efficiently predicting the nonlinear response histories at all points within the structures.

(6) A parameter analysis and an ablation study demonstrate the validity of the architectural design of StructGNN-N. Its performance showcases its great potential for use in the context of digital twins, providing an inspiring path for simulating diverse engineering structures with accurate and comprehensive mechanical information in real time.

CRediT authorship contribution statement

Linghan Song: Software, Data curation, Visualization, Investigation, Methodology, Conceptualization, Writing – original draft, Validation, Formal analysis. **Chen Wang:** Investigation, Writing – review & editing, Conceptualization, Methodology. **Jiansheng Fan:** Funding acquisition, Project administration, Supervision.

Declaration of competing interest

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

Acknowledgments

The authors gratefully acknowledge the financial support provided by the National Natural Science Foundation of China (52408188, 52293433, and 52121005).

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.eng.2025.07.001>.

References

- [1] Kurowski PM. Finite element analysis for design engineers. 3rd ed. Pittsburgh: SAE International; 2022.
- [2] Szabó B, Babuška I. Finite element analysis: method, verification and validation. 2nd ed. Hoboken: John Wiley & Sons; 2021.
- [3] Belytschko T, Liu WK, Moran B, Elkhodary K. Nonlinear finite elements for continua and structures. Hoboken: John Wiley & Sons; 2014.
- [4] Wang C, Song L, Yuan Z, Fan J. State-of-the-art AI-based computational analysis in civil engineering. *J Ind Inf Integr* 2023;3:100470.
- [5] Boje C, Guerriero A, Kubicki S, Rezgui Y. Towards a semantic construction digital twin: directions for future research. *Autom Constr* 2020;114:103179.
- [6] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521(7553):436–44.
- [7] Bengio Y, Goodfellow I, Courville A. Deep learning. Cambridge: MIT press; 2017.
- [8] Bishop CM, Bishop H. Deep learning: foundations and concepts. Cham: Springer International Publishing; 2024.
- [9] Siddique N, Adeli H. Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing. Hoboken: John Wiley & Sons; 2013.
- [10] Vadyala SR, Betgeri SN, Matthews JC, Matthews E. A review of physics-based machine learning in civil engineering. *Result Eng* 2022;13:100316.
- [11] Tapeh ATG, Naser MZ. Artificial intelligence, machine learning, and deep learning in structural engineering: a scientometrics review of trends and best practices. *Arch Comput Meth Eng* 2023;30(1):115–59.
- [12] Tao F, Liu X, Du H, Yu W. Finite element coupled positive definite deep neural networks mechanics system for constitutive modeling of composites. *Comput Meth in Appl Mech Eng* 2022;391:114548.
- [13] Rahman J, Ahmed KS, Khan NI, Islam K, Mangalathu S. Data-driven shear strength prediction of steel fiber reinforced concrete beams using machine learning approach. *Eng Struct* 2021;233:111743.
- [14] Abambres M, Lantsoght EOL. Neural network-based formula for shear capacity prediction of one-way slabs under concentrated loads. *Eng Struct* 2020;211:110501.

- [15] Feng DC, Cetiner B, Azadi Kakavand MR, Taciroglu E. Data-driven approach to predict the plastic hinge length of reinforced concrete columns and its application. *J Struct Eng* 2021;147(2):04020332.
- [16] Huang P, Chen Z, Liu Z. Nonparametric probabilistic seismic demand model and fragility analysis of subway stations using deep learning techniques. *Undergr Space* 2023;11:63–80.
- [17] Ning C, Xie Y, Sun L. LSTM, WaveNet, and 2D CNN for nonlinear time history prediction of seismic responses. *Eng Struct* 2023;286:116083.
- [18] Zhou Y, Meng S, Lou Y, Kong Q. Physics-informed deep learning-based real-time structural response prediction method. *Engineering* 2024;35:140–57.
- [19] Fei Y, Liao W, Zhao P, Lu X, Guan H. Hybrid surrogate model combining physics and data for seismic drift estimation of shear-wall structures. *Earthq Eng Struct Dyn* 2024;53(10):3093–112.
- [20] Song LH, Wang C, Fan JS, Lu HM. Elastic structural analysis based on graph neural network without labeled data. *Compos A Appl Sci Manuf* 2023;38(10):1307–23.
- [21] Wang X, Bo D, Shi C, Fan S, Ye Y, Yu PS. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Trans Big Data* 2022;9(2):415–36.
- [22] Zhao J, Wang X, Shi C, Hu B, Song G, Ye Y. Heterogeneous graph structure learning for graph neural networks. *Proc Int AAAI Conf* 2021;35(5):4697–705.
- [23] Hu Z, Dong Y, Wang K, Sun Y. Heterogeneous graph transformer. 2020. arXiv:2003.01332.
- [24] Mo X, Yang X, Chen L. Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction. 2021. arXiv:2106.07161.
- [25] Fan S, Liu G, Li J. A heterogeneous graph neural network with attribute enhancement and structure-aware attention. *IEEE Trans Comput Soc* 2023;11(1):829–38.
- [26] Fu X, King I. MECCH: metapath context convolution-based heterogeneous graph neural networks. *Neural Netw* 2024;170:266–75.
- [27] Cai D, Qian S, Fang Q, Hu J, Ding W, Xu C. Heterogeneous graph contrastive learning network for personalized micro-video recommendation. *IEEE Trans Multimed* 2022;25:2761–73.
- [28] Chen M, Huang C, Xia L, Wei W, Xu Y, Luo R. Heterogeneous graph contrastive learning for recommendation. 2023. arXiv:2303.00995.
- [29] Ma Y, Yan N, Li J, Mortazavi M, Chawla NV. HetGPT: harnessing the power of prompt tuning in pre-trained heterogeneous graph neural networks. In: *Proceedings of the ACM on Web Conference 2024*; 2024 May 13–17; Singapore, Singapore. New York City: Association for Computing Machinery; 2024. p. 1015–23.
- [30] Zhao Y, Li H, Zhou H, Attar HR, Pfaff T, Li N. A review of graph neural network applications in mechanics-related domains. *Artif Intell Rev* 2024;57(11):315.
- [31] Black N, Najafi AR. Learning finite element convergence with the multi-fidelity graph neural network. *Comput Meth Appl Mech Eng* 2022;397:115120.
- [32] Li Q, Wang Z, Li L, Hao H, Chen W, Shao Y. Machine learning prediction of structural dynamic responses using graph neural networks. *Comput Struct* 2023;289:107188.
- [33] Xue T, Adriaenssens S, Mao S. Learning the nonlinear dynamics of mechanical metamaterials with graph networks. *Int J Mech Sci* 2023;238:107835.
- [34] Yacouti M, Shakiba M. Integrated convolutional and graph neural networks for predicting mechanical fields in composite microstructures. *Compos A Appl Sci Manuf* 2025;190:108618.
- [35] Hendriks F, Menkovski V, Doškář M, Geers MGD, Rokoš O. Similarity equivariant graph neural networks for homogenization of metamaterials. *Compos A Appl Sci Manuf* 2025;439:117867.
- [36] Deshpande S, Bordas S, Lengiewicz J. Magnet: a graph u-net architecture for mesh-based simulations. 2022. arXiv:2211.00713.
- [37] Tian Y, Guan X, Sun H, Bao Y. An adaptive structural dominant failure modes searching method based on graph neural network. *Reliab Eng Syst Saf* 2024;243:109841.
- [38] Chen Q, Cao J, Lin W, Zhu S, Wang S. Predicting dynamic responses of continuous deformable bodies: a graph-based learning approach. *Compos A Appl Sci Manuf* 2024;420:116669.
- [39] Zhang C, Tao MX, Wang C, Fan JS. End-to-end generation of structural topology for complex architectural layouts with graph neural networks. *Compos A Appl Sci Manuf* 2024;39(5):756–75.
- [40] Zhang C, Tao M, Wang C, Yang C, Fan J. Differentiable automatic structural optimization using graph deep learning. *Adv Eng Inform* 2024;60:102363.
- [41] Zhao P, Liao W, Huang Y, Lu X. Beam layout design of shear wall structures based on graph neural networks. *Autom Constr* 2024;158:105223.
- [42] Fei Y, Liao W, Lu X, Guan H. Knowledge-enhanced graph neural networks for construction material quantity estimation of reinforced concrete buildings. *Compos A Appl Sci Manuf* 2024;39(4):518–38.
- [43] Zhang P, Zhao H, Shao Z, Xie X, Hu H, Zeng Y, et al. Enhanced multi-scenario running safety assessment of railway bridges based on graph neural networks with self-evolutionary capability. *Eng Struct* 2024;319:118785.
- [44] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 2021;32(1):4–24.
- [45] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, et al. Graph neural networks: a review of methods and applications. *AI open* 2020;1:57–81.
- [46] Wang C, Song L, Fan J. End-to-end structural analysis in civil engineering based on deep learning. *Autom Constr* 2022;138:104255.
- [47] Wang C, Xu L, Fan J. A general deep learning framework for history-dependent response prediction based on UA-Seq2Seq model. *Comput Meth Appl Mech Eng* 2020;372:113357.
- [48] Yousuf H, Lahzi M, Salloum SA, Shaalan K. A systematic review on sequence-to-sequence learning with neural network and its models. *Int J Electr Comput Eng* 2021;11(3):2315–26.
- [49] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys D* 2020;404:132306.
- [50] Xu L, Nie X, Fan J, Tao M, Ding R. Cyclic hardening and softening behavior of the low yield point steel BLY160: experimental response and constitutive modeling. *Int J Plast* 2016;78:44–63.
- [51] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. 2014. arXiv:1409.3215.
- [52] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. 2017. arXiv:1706.03762.
- [53] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? 2018. arXiv:1810.00826.
- [54] Baltrušaitis T, Ahuja C, Morency LP. Multimodal machine learning: a survey and taxonomy. *IEEE Trans Pattern Anal Mach Intell* 2019;41(2):423–43.
- [55] Choromanski K, Likhoshesterov V, Dohan D, Song X, Gane A, Sarlos T, et al. Rethinking attention with performers. 2020. arXiv:2009.14794.
- [56] Yoon S. Virtual sensing in intelligent buildings and digitalization. *Autom Constr* 2022;143:104578.
- [57] Pacific Earthquake Engineering Research Center. PEER ground motion database [Internet]. Berkeley: University of California; 2024 Jan 1 [cited 2024 Feb 5]. Available from: <https://ngawest2.berkeley.edu>.
- [58] Javanbakht Z, Öchsner A. Advanced finite element simulation with MSC Marc. Cham: Springer International Publishing; 2017.
- [59] Öchsner A, Öchsner M. A first introduction to the finite element analysis program MSC Marc/Mentat. Cham: Springer International Publishing; 2018.
- [60] Ba JL, Kiros JR, Hinton GE. Layer normalization. 2016. arXiv:1607.06450.
- [61] Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv:1412.6980.
- [62] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. 2019. arXiv:1912.01703.
- [63] Stevens E, Antiga L, Viehmann T. Deep learning with PyTorch. New York City: Manning Publications; 2020.
- [64] Yao W, Bai J, Liao W, Chen Y, Liu M, Xie Y. From CNN to transformer: a review of medical image segmentation models. *J Imaging Inform Med* 2024;37:1529–47.
- [65] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. 2014. arXiv:1406.1078.
- [66] Sony S, Dunphy K, Sadhu A, Capretz M. A systematic review of convolutional neural network-based structural condition assessment techniques. *Eng Struct* 2021;226:111347.
- [67] Tao F, Zhang H, Zhang C. Advancements and challenges of digital twins in industry. *Nat Comput Sci* 2024;4(3):169–77.
- [68] Javid M, Haleem A, Suman R. Digital twin applications toward industry 4.0: a review. *Cogn Robot* 2023;3:71–92.
- [69] Alnowaiser KK, Ahmed MA. Digital twin: current research trends and future directions. *Arab J Sci Eng* 2023;48(2):1075–95.