

Article

Mechanically Constrained Graph Neural Networks for Linear Static Analysis of Planar Frame Structures with Member Loads

Kun Leng ¹ , Peng He ^{1,*}, Huijun Jiang ¹, Qiang Feng ¹, Wen Zheng ² and Yu Zhou ¹

¹ School of Civil Engineering and Architecture, Shandong University of Science and Technology, Qingdao 266000, China; lengkun2024@sdust.edu.cn (K.L.); jianghuijun@sdust.edu.cn (H.J.); fengqiang@sdust.edu.cn (Q.F.); zhouyu2025@sdust.edu.cn (Y.Z.)

² Fujian Huichuan IoT Technology Co., Ltd., Fuzhou 350000, China; z0161@sdust.edu.cn

* Correspondence: hepenghank1@sdust.edu.cn

Abstract

Artificial intelligence methods have attracted increasing interest in structural analysis. However, engineering applications still require reliable representation of irregular topology, actual member loads, mechanical consistency, and interpretable structural response quantities. This study develops a mechanically constrained graph neural network (GNN) method for 2D linear elastic static analysis of planar truss and building frame structures. The method represents the structural system as a graph and encodes nodes, members, boundary conditions, nodal loads, member stiffness, and actual member loads as graph features. A message-passing GNN predicts nodal DOF displacements, and the loss function includes free DOF equilibrium residuals and global equilibrium residuals to improve mechanical consistency. The workflow then recovers support reactions and member axial forces, shear forces, and bending moments from the predicted displacements through structural mechanics relations. Numerical examples show that the proposed method reduces equilibrium residuals and improves member force recovery compared with a displacement-supervised baseline. Ablation studies further confirm the roles of member load input, equilibrium constraints, and internal force recovery. The results show that graph representation, training constrained by equilibrium, and mechanics-based recovery can be integrated into an interpretable framework for structural static analysis.

Keywords: graph neural network; mechanical constraint; structural static analysis; actual member load; internal force recovery



Academic Editor: Huiyong Ban

Received: 9 May 2026

Revised: 21 June 2026

Accepted: 23 June 2026

Published: 25 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Structural static analysis is central to structural design, performance assessment, and engineering safety evaluation. Conventional finite element and stiffness method solvers accurately compute nodal displacements, support reactions, and member internal forces for linear elastic static problems, and they remain essential in engineering practice. However, repeated design exploration, studies involving many load cases, and learning-based workflows raise additional requirements. A model must represent irregular structural topology, describe loads applied directly to members, preserve mechanical consistency, and recover engineering quantities such as support reactions and member internal forces. In recent years, researchers have increasingly introduced artificial intelligence and machine learning methods into structural engineering, applying them to structural response prediction, performance assessment, reliability analysis, and design support. Existing

reviews show that the application scope of artificial intelligence in structural engineering continues to expand, with machine learning gradually extending from material-property prediction and member-capacity assessment to structural-system-level response analysis and design evaluation [1–5]. These studies provide a basis for developing rapid structural analysis models. Nevertheless, structural systems are characterized by irregular topologies, diverse boundary conditions, complex member properties, and non-uniform loading forms. Directly representing structural systems with fixed-length vectors or regular grids often fails to preserve the connectivity between structural nodes and members.

Researchers have also used deep learning surrogate models to predict response fields in structural members and continuum mechanics problems. Previous studies have attempted to use deep networks to predict stress distributions, stress concentrations in damaged members, transient finite element results, and parametric structural mechanics responses [6–10]. Finite-element-integrated neural networks, recurrent neural networks, and mesh-free surrogate models have also been used to accelerate multiscale mechanical simulations or structural mechanics finite element simulations [11–13]. These methods demonstrate the potential of neural networks to learn complex mappings between input parameters and mechanical responses. However, many models still mainly rely on regular-grid inputs or fixed-dimensional parameter representations, making them difficult to apply to structural systems with different numbers of nodes, member connection patterns, and topologies. In addition, model outputs usually focus on displacements, stresses, or response fields, while member internal forces, support reactions, and their sign conventions, which are commonly used in engineering static analysis, receive insufficient attention.

Physics-informed machine learning offers one way to address the limitations of models driven purely by data. PINNs and related scientific machine learning methods incorporate governing equations, boundary conditions, or energy functionals into the loss function. As a result, training need not rely entirely on large labeled datasets, and predictions can better satisfy mechanical constraints [14–16]. In solid mechanics, physics-informed deep learning has been applied to inverse modeling, linear elasticity, exact boundary condition treatment, multi-material problems, 2D and 3D solid mechanics computation, hyperelastic analysis, nonlocal mechanics, and deep energy methods [17–24]. In addition, physics-informed DeepONets and unlabeled elastodynamic models indicate that neural networks can learn solution operators or dynamic responses for parametric problems under physical-equation constraints [25,26]. These studies show that incorporating physical equations into training helps alleviate data scarcity. However, most methods still focus on continuous domains, regular grids, or material-mechanics problems, leaving room for further research on graph representation, DOF-level equilibrium constraints, and engineering internal force recovery for node-member structural systems. For structural static analysis, this issue is particularly important because a small displacement prediction error does not necessarily guarantee static equilibrium or reliable member internal forces. Therefore, a learning-based structural analysis model should be evaluated not only by displacement accuracy, but also by mechanical consistency and its ability to produce engineering-interpretable quantities.

Graph neural networks provide a natural data structure for modeling structural systems. In a structural system, nodes correspond to joints or DOF locations, whereas members correspond to beams, columns, braces, and other connections. This correspondence aligns directly with nodes and edges in graph data. Song et al. proposed a label-free GNN model for elastic structural analysis, demonstrating that structural topology, member attributes, and mechanical equations can be incorporated within a graph-learning framework [27]. StructGNN further verified the computational efficiency of GNNs in static structural analysis [28]. In structural engineering systems, mechanics-informed GNN models have been used for structural response analysis, and GNNs have also been applied to structural

dynamic response prediction and accelerated numerical simulation [29–31]. More broadly in mechanics, GNNs can predict stress, strain, and deformation fields in materials and structures, and have also been used for stiffened-panel stress prediction, energy-informed graph transformers, physical system surrogate modeling, graph neural Galerkin networks, and soft-tissue mechanics simulation [32–37]. Collectively, these studies indicate that GNNs can represent non-Euclidean structural topologies and transmit mechanical response information along local connections, making them suitable for structural analysis tasks involving node-member systems. Although these studies have established important foundations for graph-based structural and mechanics analysis, their research emphases differ in structural analysis tasks, mechanical constraints, and engineering output forms. To clarify the relationship between the present work and representative existing studies, Table 1 summarizes their main contributions and research positioning.

Table 1. Representative physics-informed and graph-based structural and mechanics studies.

Study	Main Contribution	Research Focus
PINNs and physics-informed learning methods	Incorporate governing equations, boundary conditions, or energy functionals into neural-network training and have been applied to solid mechanics, elasticity, hyperelasticity, and elastodynamics.	Physics-informed learning for continuum, solid mechanics, and operator learning problems.
Song et al. [27]	Proposed a label-free GNN framework for elastic structural analysis and showed that structural topology, member attributes, and mechanical equations can be incorporated into graph learning.	Physics-constrained GNN for elastic structural systems.
Chou et al. [28]	Developed an efficient GNN framework for static structural analysis and demonstrated the potential of GNNs for structural response prediction.	GNN-based response prediction framework for static structural analysis.
Fang et al. [29].	Proposed a mechanics-informed GNN method that incorporates structural mechanics knowledge into graph learning for structural response prediction.	Mechanics-informed graph learning for structural response prediction.
Li et al. [30]	Applied GNNs to structural dynamic response prediction and demonstrated the ability of graph learning to model time-dependent structural responses.	Graph-learning prediction of structural dynamic responses.
Other graph-based mechanics studies [31–37]	Used graph, mesh, or material point representations for stress, strain, deformation field prediction, energy-informed modeling, physical system surrogate modeling, graph neural Galerkin methods, and soft-tissue mechanics simulation.	Graph learning for continuum, material, and mechanics field prediction problems.
Present study	Develops a mechanically constrained GNN framework for 2D linear elastic static analysis of planar truss and frame structures with actual member loads. The method integrates node-member structural graph representation, actual member load encoding, free DOF and global equilibrium constraints, and mechanics-based recovery of support reactions and member axial forces, shear forces, and bending moments into a unified workflow.	Mechanically interpretable node-member graph-learning framework for engineering static-analysis outputs.

Table 1 is intended to clarify methodological positioning, not to claim general superiority over existing PINN, Graph Transformer, or GNN-based methods. These methods often differ in data representation, target outputs, mechanical constraints, and intended structural analysis tasks. Overall, existing studies have demonstrated the feasibility of GNNs for structural and mechanics problems from the perspectives of elastic structural graph

learning, static response prediction, mechanics-informed constraints, dynamic response prediction, and mechanics field prediction. Building on these contributions, the present study focuses on a workflow for engineering static analysis that integrates actual member load input, displacement prediction constrained by equilibrium, and mechanics-based recovery of support reactions and member axial forces, shear forces, and bending moments.

Despite these advances, existing GNN-based structural analysis studies have not fully addressed several engineering requirements at the same time. First, many models focus on nodal displacement, stress, or response field prediction. They address support reactions and member internal forces less explicitly, although these quantities are standard outputs in engineering static analysis. Second, external loads are often represented as nodal loads, global descriptors, or other simplified input quantities, whereas practical frame and truss structures may be subjected to distributed or concentrated loads acting directly on members. If such member loads are not explicitly encoded in the graph, the structural loading state received by the model remains incomplete. Third, engineering-oriented structural analysis requires not only small prediction errors but also satisfaction of static equilibrium, boundary conditions, and local-coordinate internal force conventions. Data-driven mechanics and constitutive-modeling studies have shown that learned mechanical models should satisfy mechanical admissibility and interpretability requirements [38–40]. In addition, structural reliability and optimization studies indicate that reliable response models are valuable for repeated analyses such as load-case evaluation and design iteration [41–43], while recent physics-informed solid and structural mechanics studies further show that boundary treatment, loss-function construction, and engineering applicability remain key challenges [44,45]. Therefore, this study does not address a shortage of efficient solvers for linear elastic static equilibrium equations. Rather, it focuses on the need for an engineering-oriented graph learning workflow. Such a workflow should represent node-member topology, encode actual member loads, incorporate static equilibrium constraints during training, and recover support reactions and member internal forces from predicted displacement responses.

This study does not aim to replace conventional finite element or stiffness method solvers for a single deterministic linear elastic static analysis. Such solvers remain accurate, efficient, and essential in routine engineering design. The present work develops a mechanically constrained graph neural network framework for node-member structural systems, where structural topology, boundary conditions, member stiffness, nodal loads, and actual member loads are represented in a unified graph format. The mechanical constraints and mechanics-based recovery procedure connect the learned displacement response to engineering static analysis quantities such as support reactions and member internal forces.

Against this background, this study develops a mechanically constrained GNN framework for 2D linear elastic static analysis of planar truss and frame structures. Planar truss and frame structures are represented as structural graphs, and nodal coordinates, boundary conditions, nodal loads, member geometry and stiffness parameters, and actual member loads are encoded as node and edge features. A GNN then performs message passing on the structural graph to predict nodal DOF displacements. During training, free DOF static equilibrium residuals and global equilibrium residuals are introduced into the loss function to constrain the predicted displacements to satisfy structural mechanics relations. Finally, support reactions and member P , V_2 , and M_3 internal force results are recovered from the predicted displacements and local member mechanics relations. Unlike models that only fit nodal displacements, the proposed method emphasizes a complete workflow from graph input and displacement prediction to equilibrium constraints and engineering internal force recovery.

2. Graph Representation of the Structural Static Analysis Problem

Structural static analysis solves for nodal displacements, support reactions, and member internal forces under specified geometry, material and section parameters, boundary conditions, and external loads. For planar truss and frame structures, the structural system consists of discrete nodes and bars or beam-column members connecting those nodes. Its mechanical behavior depends not only on the stiffness parameters of individual members, but also on nodal connectivity, constraints, and load transfer paths within the structural system. Therefore, the static response of a structure is inherently topology-dependent. Conventional finite element methods usually describe this problem through node numbering, element connectivity, and assembly of the global stiffness matrix. From a data-structure perspective, this procedure naturally corresponds to nodes, edges, and adjacency relationships in a graph.

To support graph neural network modeling of non-Euclidean data, planar truss and frame structures are represented as structural graphs:

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

where $\mathcal{V} = \{v_i\}_{i=1}^N$ is the set of structural nodes and N is the number of nodes; $\mathcal{E} = \{e_{ij}\}_{j=1}^M$ is the set of members and M is the number of members. Each node v_i corresponds to a physical node in the structure, and each edge e_{ij} corresponds to the structural member connecting node i and node j . Unlike regular image grids, the number of nodes, connection patterns, and member orientations in structural systems usually vary with the engineering object. Therefore, data representations that rely on regular grids, such as convolutional neural networks, are difficult to apply directly. Graph representation can describe different structural forms without altering the physical topology, thereby providing a data basis for subsequent GNN-based learning of structural responses. The mapping from a structural system to graph data is shown in Figure 1.

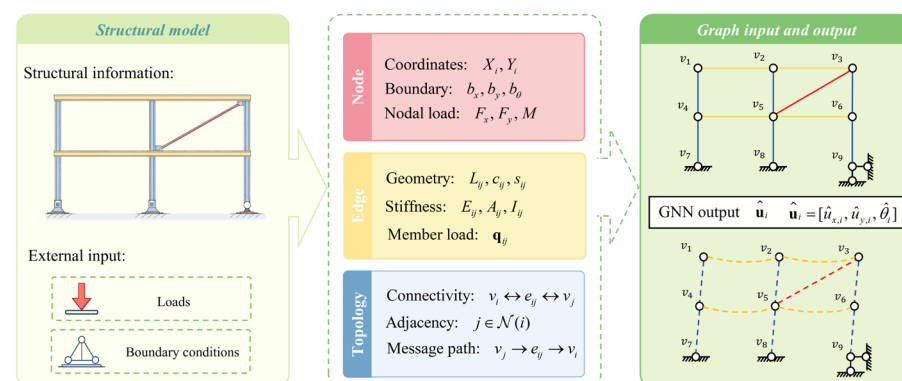


Figure 1. Mapping from a structural system to graph data.

For a planar frame structure, each node usually contains three degrees of freedom, namely horizontal displacement, vertical displacement, and rotation:

$$\mathbf{u}_i = [u_{x,i}, u_{y,i}, \theta_i]^T \quad (2)$$

The global nodal displacement vector can be expressed as

$$\mathbf{u} = [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_N^T]^T \quad (3)$$

The proposed GNN model takes the structural graph G as input and uses the displacement of each nodal DOF as the primary prediction target. In other words, the model first

learns the mapping from structural topology, member attributes, boundary conditions, and load information to nodal displacement response:

$$\hat{\mathbf{u}} = f_{\theta}(G) \quad (4)$$

where $f_{\theta}(\cdot)$ denotes the graph neural network model parameterized by θ , and $\hat{\mathbf{u}}$ is the predicted nodal displacement. Instead of directly predicting member internal forces, this study uses nodal displacement as the basic GNN output because displacement is the core unknown in the global stiffness relation of a structure. Once nodal displacements are obtained, support reactions and member-end forces can be further recovered from member stiffness relations, thereby forming a complete workflow from graph input to engineering static-analysis results.

At the node level, a node feature vector is constructed for each structural node. The node features include not only geometric position information, but also boundary and load information relevant to static analysis, which can be written as $\mathbf{x}_i$:

$$\mathbf{x}_i = [X_i, Y_i, \mathbf{b}_i, \mathbf{p}_i, \mathbf{d}_i] \quad (5)$$

where (X_i, Y_i) denotes the nodal coordinates; $\mathbf{b}_i$ denotes the nodal boundary constraint state and describes whether the horizontal displacement, vertical displacement, and rotational DOFs are constrained; $\mathbf{p}_i$ denotes the external load applied to the node, including nodal horizontal force, vertical force, and moment; and $\mathbf{d}_i$ can denote the nodal degree or other auxiliary information reflecting the local topological state. Introducing boundary constraint information enables the GNN to distinguish free nodes from support nodes, thereby avoiding treating all nodes as the same type of prediction object. Introducing nodal load information enables the model to perceive the location and direction of external actions in the structural system.

At the member level, each edge corresponds to a structural member, and its edge feature vector describes member geometry, material, section, and load attributes. For a planar frame member, the edge features can be expressed as e_{ij} $\mathbf{a}_{ij}$:

$$\mathbf{a}_{ij} = [L_{ij}, c_{ij}, s_{ij}, E_{ij}, A_{ij}, I_{ij}, \mathbf{q}_{ij}] \quad (6)$$

where L_{ij} is the member length; c_{ij} and s_{ij} are the direction cosines of the member local axis relative to the global coordinate system; E_{ij} is the elastic modulus; A_{ij} is the cross-sectional area; I_{ij} is the second moment of area; and $\mathbf{q}_{ij}$ denotes the actual external action applied to the member. The member load information is not limited to uniformly distributed loads; it may also include concentrated loads, linearly varying loads, or other parameterized actions applied at the member level. By incorporating actual member loads into the edge features, the structural graph contains not only geometry and stiffness information, but also the loading information commonly used in engineering analysis. For clarity, the main graph input features used in this study and their physical meanings are summarized in Table 2.

Based on the node and edge features described above, the adjacency relationship of the structural graph is determined by member connectivity. If a member exists between node i and node j , an edge e_{ij} exists in the graph. Because frame members have two clearly defined end nodes, edge features naturally contain member orientation, length, and local-coordinate information. For an undirected structural connection, the member can be treated as a bidirectional edge during GNN message passing, so that node i can receive information from node j and node j can also receive information from node i . In this way, load, boundary, and stiffness information in the structure can be transmitted along member connections on the graph.

Table 2. Graph input features and physical meanings.

Feature Category	Symbol	Physical Meaning	Object
Nodal coordinates	(X_i, Y_i)	Spatial position of node	Node
Boundary constraint	$\mathbf{b}_i$	Constraint state of each DOF	Node
Nodal load	$\mathbf{p}_i$	Nodal forces and nodal moments	Node
Member length	L_{ij}	Member geometric scale	Edge
Direction cosines	c_{ij}, s_{ij}	Direction of local axis	Edge
Material parameter	E_{ij}	Elastic modulus	Edge
Section parameters	A_{ij}, I_{ij}	Parameters related to axial and flexural stiffness	Edge
Member load	$\mathbf{q}_{ij}$	External action on member	Edge

This graph representation transforms structural static analysis into a graph-to-nodal-response prediction problem. Nodes and members correspond to graph nodes and edges, respectively, and structural connectivity defines the message-passing paths. Nodal coordinates, boundary conditions, nodal loads, member geometry and stiffness parameters, and actual member loads jointly constitute the model input. This representation preserves the irregular topology of the structural system while enabling the GNN to perceive structural geometry, constraints, stiffness, and external actions within a unified data format.

3. GNN Static Analysis Model with Mechanical Constraints

Using the structural graph representation developed in Section 2, this study constructs a GNN model for displacement prediction in engineering static analysis. The model takes the structural graph $G = (\mathcal{V}, \mathcal{E})$ as input, learns the coupled relationships among nodes, members, boundary conditions, and external actions through graph message passing, and outputs structural nodal DOF displacements. Unlike supervised learning models based only on label fitting, the proposed model embeds the structural static equilibrium equation into the loss function, so that training is constrained by mechanical equations and the dependence on large amounts of displacement labels is reduced.

3.1. Overall Framework for Engineering Static Analysis

The overall workflow contains four components: structural graph input, GNN displacement prediction, training constrained by mechanical equations, and member internal force recovery. First, a structural graph is constructed from structural nodes, member connectivity, material and section parameters, boundary conditions, nodal loads, and actual member loads. The GNN performs message passing on the node-member topology, obtains nodal hidden vectors containing local member information and neighborhood topological information, and outputs the displacement of each nodal DOF. Finally, the predicted displacements are substituted into structural equilibrium relations to compute mechanical residuals and further recover member internal force responses.

To clarify the relationships among these components, Figure 2 presents the proposed mechanically constrained GNN framework for static analysis. The structural graph input module organizes structural topology, node features, and edge features. The GNN message-passing module learns response-transfer relationships in the node-member system. The displacement prediction module provides the direct neural-network output. The mechanical constraint module introduces the static equilibrium equation into the loss function and constrains network training through backpropagation. The internal force recovery module obtains member internal force responses from the predicted displacements and member mechanical relations.

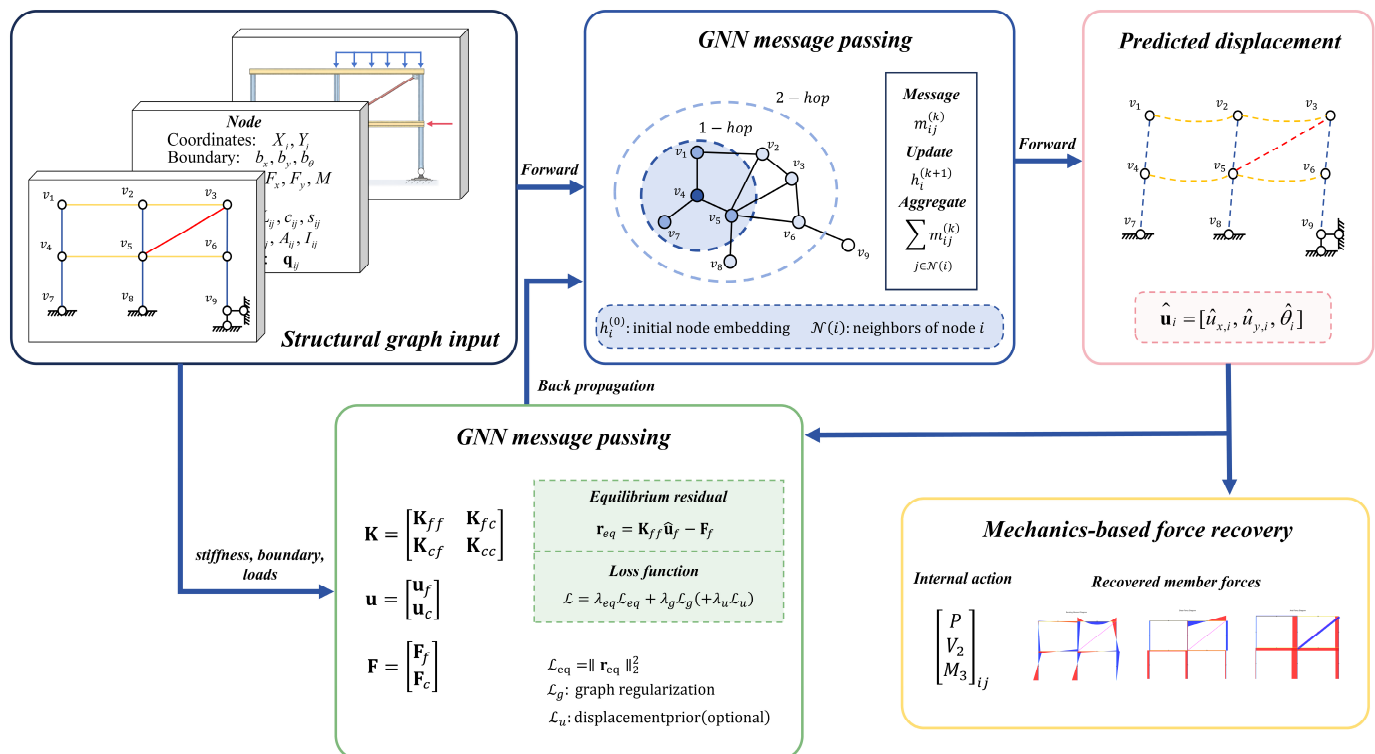


Figure 2. Mechanically constrained GNN framework for static analysis.

As shown in the figure, displacement prediction and internal force recovery have different meanings in the proposed method. The nodal displacement $\hat{\mathbf{u}}_i$ is the direct output of the GNN, whereas the member-end force $\hat{\mathbf{f}}_e$ is not a direct black-box network prediction; it is recovered by combining the predicted displacement with local member mechanics relations. At the same time, the equilibrium residual $\mathbf{r}_{eq}$ is introduced into the loss function to constrain the predicted displacements to satisfy static equilibrium during training. Therefore, the proposed framework includes both a data-driven graph-learning process and a mechanical-equation constraint process.

The workflow can be expressed as

$$G = (\mathcal{V}, \mathcal{E}) \rightarrow \mathbf{h}_i \rightarrow \hat{\mathbf{u}}_i \rightarrow \{\mathbf{r}_{eq}, \hat{\mathbf{f}}_e\} \quad (7)$$

where $\mathbf{h}_i$ is the hidden vector of node i , $\hat{\mathbf{u}}_i$ is the nodal displacement predicted by the GNN, $\mathbf{r}_{eq}$ is the static equilibrium residual, and $\hat{\mathbf{f}}_e$ is the member-end force recovered from the predicted displacement. Here, $\hat{\mathbf{u}}_i$ is the direct output of the neural network, whereas $\hat{\mathbf{f}}_e$ is a derived response obtained from the predicted displacement and structural mechanics relations.

3.2. Message Passing and Displacement Prediction on the Structural Graph

Based on the structural graph $G = (\mathcal{V}, \mathcal{E})$ constructed above, a nodal displacement prediction model is further established. The model takes node features, edge features, and topological connectivity as inputs, updates nodal hidden vectors through a message-passing mechanism, and learns the mapping from the structural graph to nodal displacement responses.

During message passing, the node and edge features are first mapped into a hidden space:

$$\mathbf{h}_i^{(0)} = \phi_x(\mathbf{x}_i) \quad (8)$$

$$\mathbf{e}_{ij} = \phi_e(\mathbf{a}_{ij}) \quad (9)$$

where $\mathbf{h}_i^{(0)}$ is the initial hidden vector of node v_i , $\mathbf{e}_{ij}$ is the hidden vector of member edge e_{ij} , and $\phi_x(\cdot)$ and $\phi_e(\cdot)$ are learnable feature-mapping functions. Then, in the l -th message-passing layer, node v_i receives messages from its neighboring nodes $v_j \in \mathcal{N}(i)$ and connected members $\mathbf{e}_{ij}$:

$$\mathbf{m}_{j \rightarrow i}^{(l)} = \phi_m^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{e}_{ij}) \quad (10)$$

The neighborhood messages of a node are summarized by a permutation-invariant aggregation function v_i :

$$\bar{\mathbf{m}}_i^{(l)} = \text{AGG}\left(\left\{\mathbf{m}_{j \rightarrow i}^{(l)} \mid j \in \mathcal{N}(i)\right\}\right) \quad (11)$$

where $\text{AGG}(\cdot)$ can be summation, averaging, or attention-weighted aggregation. After message aggregation, the nodal hidden vector is updated as

$$\mathbf{h}_i^{(l+1)} = \phi_u^{(l)}(\mathbf{h}_i^{(l)}, \bar{\mathbf{m}}_i^{(l)}) \quad (12)$$

After multiple message-passing layers, the hidden vector of a node v_i contains both local member information and structural topology information within a certain range. For frame structures, multilayer message passing can be interpreted as a process in which nodal response information propagates step by step along beams, columns, or braces. As the network depth $\mathbf{h}_i^{(L)}$ increases, the model can extend from local connectivity to a broader structural load transfer path, thereby learning the combined influence of global stiffness distribution, boundary constraints, and loads on nodal displacement.

Finally, the GNN predicts nodal DOF displacements through an output function:

$$\tilde{\mathbf{u}}_i = \phi_o(\mathbf{h}_i^{(L)}) \quad (13)$$

$$\tilde{\mathbf{u}}_i = [\tilde{u}_{x,i}, \tilde{u}_{y,i}, \tilde{\theta}_i]^T \quad (14)$$

v_i denotes the raw prediction of the three nodal DOF displacements by the network. To ensure that boundary conditions are satisfied at the output level, a DOF mask is further introduced:

$$\hat{\mathbf{u}}_i = \mathbf{m}_i^f \odot \tilde{\mathbf{u}}_i + \mathbf{m}_i^c \odot \bar{\mathbf{u}}_i \quad (15)$$

where $\tilde{\mathbf{u}}_i$ is the raw network output for the three DOFs of node i , $\hat{\mathbf{u}}_i$ is the final predicted displacement after applying boundary conditions, $\mathbf{m}_i^f$ is the free DOF mask, $\mathbf{m}_i^c$ is the constrained-DOF mask, $\bar{\mathbf{u}}_i$ is the prescribed displacement of constrained DOFs, and $\odot$ denotes element-wise multiplication. For common fixed, pinned, or roller supports, the prescribed displacement of constrained DOFs is usually zero. This treatment allows the model to perform effective displacement prediction only for free DOFs while maintaining consistency between support constraints and the structural analysis problem.

With this treatment, structural topology, boundary constraints, nodal loads, member stiffness, and actual member loads are encoded together in the structural graph. The direct output of the GNN is nodal displacement. Member internal forces and support reactions are not direct network predictions; subsequent mechanical relations recover them from the predicted displacements. This setting keeps the graph learning model consistent with the structural mechanics analysis workflow and provides a unified interface for subsequent equilibrium residual constraints and engineering internal force recovery.

3.3. Training Loss Function Incorporating Physical Equations

To give the nodal displacement predictions a clear static mechanics meaning, this study introduces the structural equilibrium equation into the GNN training objective. Unlike training strategies that minimize prediction error only through displacement labels, this study constructs static equilibrium residuals on free DOFs and includes them in the loss function for backpropagation. In this way, the network output is encouraged not only to approach the reference displacement results, but also to satisfy the equilibrium relations required in structural static analysis.

For planar truss and frame structures, the discretized global static equilibrium equation can be written as

$$\mathbf{K}\mathbf{u} = \mathbf{F} \quad (16)$$

where $\mathbf{K}$ is the global stiffness matrix of the structure, $\mathbf{u}$ is the nodal DOF displacement vector, and $\mathbf{F}$ is the external action vector. This study does not use the GNN as a replacement solver for the conventional stiffness method; instead, the equilibrium relation is used to physically constrain the GNN-predicted displacement. Let the nodal displacement vector output by the GNN be $\hat{\mathbf{u}}_i$; it can then be substituted into the structural equilibrium equation to compute the residual.

Because support constraints exist in the structure, not all DOFs are suitable for directly imposing zero-residual constraints. Support reactions usually occur at constrained DOFs. If the residual is computed over all DOFs $\mathbf{K}\hat{\mathbf{u}} - \mathbf{F}$, the training process may treat support reactions as equilibrium errors and distort the loss. Therefore, this study establishes a DOF mask according to the boundary conditions, divides nodal DOFs into free and constrained DOFs, and constructs equilibrium residuals only on the free DOFs.

Let the predicted displacement corresponding to the free DOFs be $\hat{\mathbf{u}}_f$ and the prescribed displacement corresponding to the constrained DOFs be $\bar{\mathbf{u}}_c$. The static equilibrium residual on the free DOFs can then be written as

$$\mathbf{r}_{eq} = \mathbf{K}_{ff}\hat{\mathbf{u}}_f + \mathbf{K}_{fc}\bar{\mathbf{u}}_c - \mathbf{F}_f \quad (17)$$

where $\mathbf{K}_{ff}$ and $\mathbf{K}_{fc}$ are submatrices of the global stiffness matrix associated with the free and constrained DOFs, respectively, and $\mathbf{F}_f$ is the external action vector corresponding to the free DOFs. For common zero-displacement support constraints, $\bar{\mathbf{u}}_c = 0$; the above equation can be simplified as

$$\mathbf{r}_{eq} = \mathbf{K}_{ff}\hat{\mathbf{u}}_f - \mathbf{F}_f \quad (18)$$

$$\mathbf{r}_{eq} = \mathbf{M}_f(\mathbf{K}\hat{\mathbf{u}} - \mathbf{F}) \quad (19)$$

where $\mathbf{M}_f$ is the free DOF masking matrix, which extracts the equilibrium residual corresponding to the free DOFs from the global residual. This treatment does more than identify whether a node is constrained. It applies the structural static equilibrium equation at the nodal DOF level and constrains the equilibrium state of the free DOFs.

Based on the above residual, the free DOF equilibrium loss is defined as

$$\mathcal{L}_{eq} = \frac{1}{n_f} \mathbf{r}_{eq2}^2 \quad (20)$$

where n_f is the number of free DOFs. To reduce the influence of different load magnitudes on the loss scale, a normalized residual form can also be used:

$$\mathcal{L}_{eq}^* = \frac{\mathbf{r}_{eq2}^2}{\mathbf{F}_{f2}^2 + \varepsilon} \quad (21)$$

where ε is a small positive constant used to avoid a zero denominator. The normalized equilibrium residual can be used for monitoring the training process and as an indicator for comparing mechanical consistency among different model configurations.

In addition to the free DOF equilibrium residual, a global equilibrium constraint is further introduced to constrain the loading state corresponding to the predictions at the structural-system level. Let the global force and moment equilibrium residuals be

$$\mathbf{r}_g = [r_{Fx}, r_{Fy}, r_M]^T \quad (22)$$

where r_{Fx} and r_{Fy} denote the resultant force imbalance in the global horizontal and vertical directions, respectively, and r_M denotes the global resultant moment imbalance about a selected reference point. The global equilibrium residual is computed from support reactions recovered from the predicted displacements, external nodal loads, and equivalent member loads. To avoid the influence of load magnitude on the residual scale, the resultant force residual and resultant moment residual are normalized by a reference force and a reference moment, respectively. Because the support reactions are computed from the predicted displacements through the global stiffness relation, the global equilibrium residual is differentiable with respect to the predicted displacement and can therefore be included in the network loss function. The corresponding global equilibrium loss is

$$\mathcal{L}_g = \mathbf{r}_g^2 \quad (23)$$

The free DOF equilibrium residual focuses on local static consistency at the nodal DOF level. The global equilibrium residual checks whether external actions, support reactions, and structural responses are consistent at the structural system level. The two types of equilibrium constraints work together to reduce the risk that the model performs well in displacement error but remains unreliable with respect to static equilibrium.

When displacement labels are available, a displacement supervision loss is also introduced:

$$\mathcal{L}_u = \frac{1}{n_f} \hat{\mathbf{u}}_f - \mathbf{u}_f^2 \quad (24)$$

where $\mathbf{u}_f$ is the reference displacement. The comprehensive training objective can be written as

$$\mathcal{L} = \lambda_u \mathcal{L}_u + \lambda_{eq} \mathcal{L}_{eq} + \lambda_g \mathcal{L}_g \quad (25)$$

where λ_u , λ_{eq} , and λ_g are the weighting coefficients of different loss terms. To clarify the role of each loss term in model training, Table 3 summarizes the components of the loss function and their physical meanings. The displacement supervision term improves the model's ability to fit the reference displacement response, the equilibrium residual term constrains predictions to satisfy static equilibrium at the nodal DOF level, and the global equilibrium term further constrains the overall loading state of the structure.

Table 3. Components and roles of the loss function.

Loss Term	Expression	Role
Displacement supervision loss	$\mathcal{L}_u$	Auxiliary term: constrains predicted displacements toward reference values
Free-DOF equilibrium loss	$\mathcal{L}_{eq}$	Embeds static equilibrium equations into training
Global equilibrium loss	$\mathcal{L}_g$	Constrains global force and moment equilibrium

Through the above loss-function design, GNN training is jointly constrained by displacement supervision and mechanical equilibrium residuals. When sufficient displacement

labels are available, the supervised term can improve training stability and displacement prediction accuracy. When displacement labels are insufficient, the equilibrium residual term provides a constraint signal from the structural static equation in addition to displacement supervision, helping reduce the risk that the model relies entirely on displacement fitting and making predictions closer to the equilibrium state required in engineering static analysis.

3.4. Engineering Internal-Force Recovery Based on Predicted Displacements

In the proposed workflow, the GNN directly predicts nodal DOF displacements on the structural graph. The model does not treat support reactions or member internal forces as unconstrained network outputs. Instead, deterministic structural mechanics relations recover these quantities from the predicted displacements.

Accordingly, the mechanics-based recovery step converts GNN-predicted displacements into member axial forces, shear forces, bending moments, and support reactions commonly used in structural static analysis.

For member e_{ij} , the predicted displacements of the two end nodes are first extracted from the global nodal displacement vector to form the member global displacement vector:

$$\hat{\mathbf{u}}_e^g = [\hat{u}_{x,i}, \hat{u}_{y,i}, \hat{\theta}_i, \hat{u}_{x,j}, \hat{u}_{y,j}, \hat{\theta}_j]^T \quad (26)$$

where the superscript g denotes the global coordinate system. For a planar frame member, internal forces are usually defined in the member local coordinate system. Therefore, a coordinate transformation matrix must be established according to member orientation to transform the global displacement into the local displacement $\mathbf{T}_e$:

$$\hat{\mathbf{u}}_e^l = \mathbf{T}_e \hat{\mathbf{u}}_e^g \quad (27)$$

where the superscript l denotes the member local coordinate system. The local coordinate system usually takes the member axis as the local axial direction and the direction perpendicular to the member axis as the local transverse direction. Through this transformation, member internal forces with different spatial orientations and connection forms can be expressed uniformly in their own member coordinate systems.

After the member local displacement is obtained, the member-end force can be recovered from the local stiffness relation:

$$\hat{\mathbf{f}}_e^l = \mathbf{k}_e^l \hat{\mathbf{u}}_e^l + \Delta \mathbf{f}_{q,e}^l \quad (28)$$

where $\mathbf{k}_e^l$ is the element stiffness matrix in the member local coordinate system, and $\Delta \mathbf{f}_{q,e}^l$ is the internal force correction term caused by the actual external action on the member. This correction term reflects the contribution of distributed loads, concentrated loads, or other loads applied to members to member-end forces and internal force diagrams. The signs depend on the adopted local coordinate direction and positive end-force convention, but the convention must remain consistent within each structural example.

For a planar frame member, the recovered local member-end force vector can be expressed as

$$\hat{\mathbf{f}}_e^l = [\hat{N}_i, \hat{V}_i, \hat{M}_i, \hat{N}_j, \hat{V}_j, \hat{M}_j]^T \quad (29)$$

where $\hat{N}_i$, $\hat{V}_i$, and $\hat{M}_i$ denote the member-end axial force, shear force, and bending moment, respectively. To maintain consistency with the common output form of engineering analysis, the local member-end forces are further organized into member-level P , V_2 , and M_3

responses, where P denotes member axial force, V_2 denotes local transverse shear force, and M_3 denotes bending moment about the principal axis of the local section.

If member internal force diagrams are required, section internal forces can be further calculated along the member length. In general, this can be written as $x \in [0, L_e]$:

$$\hat{\mathbf{s}}_e(x) = \mathbf{B}_e(x)\hat{\mathbf{u}}_e^l + \Delta\mathbf{s}_{q,e}(x) \quad (30)$$

where $\hat{\mathbf{s}}_e(x)$ denotes the internal force response at an arbitrary section of a member, $\mathbf{B}_e(x)$ is the recovery matrix determined by the member shape functions and stiffness relation, and $\Delta\mathbf{s}_{q,e}(x)$ is the contribution of external member actions to the section internal forces. By sampling along the member length, axial-force, shear-force, and bending-moment diagrams can be obtained.

Support reactions can also be recovered from the predicted displacements. Let the reaction vector corresponding to the constrained DOFs be $\hat{\mathbf{R}}_c$; then

$$\hat{\mathbf{R}}_c = \mathbf{K}_{cf}\hat{\mathbf{u}}_f + \mathbf{K}_{cc}\bar{\mathbf{u}}_c - \mathbf{F}_c \quad (31)$$

where $\mathbf{K}_{cf}$ and $\mathbf{K}_{cc}$ are submatrices of the global stiffness matrix associated with the constrained DOFs, and $\mathbf{F}_c$ is the external action vector corresponding to the constrained DOFs. For zero-displacement supports, $\bar{\mathbf{u}}_c = 0$. This process shows that support reactions are also not direct network outputs, but are computed from the predicted displacements and the global equilibrium relation.

In summary, this study establishes a continuous analysis process linking structural graph input, nodal displacement prediction, and member internal force recovery. The GNN represents the influence of structural topology, boundary conditions, member stiffness, and actual member loads on displacement response. The physical loss function constrains the predicted displacements to satisfy static equilibrium, and the internal force recovery process further transforms the predicted displacements into engineering responses such as support reactions and member axial forces, shear forces, and bending moments. This process extends the engineering representation of GNN displacement predictions and improves the mechanical interpretability of model outputs.

4. Numerical Tests and Results

To verify the effectiveness of the proposed mechanically constrained GNN method for static structural analysis, a planar frame structure is selected as the numerical example in this section. The analysis focuses on nodal displacement prediction, static-equilibrium consistency, and member internal force recovery. Unlike studies that only compare nodal displacement errors, this section emphasizes whether the GNN predictions can further satisfy the mechanical constraints required for structural static analysis and whether they can output engineering-meaningful member internal force responses after considering actual member load input and the engineering internal force recovery process. Specifically, the numerical tests first examine the ability of structural graph input and GNN message passing to learn the displacement response of a planar frame, then evaluate the improvement in static consistency after physical equilibrium equations are introduced into the loss function, and finally analyze the accuracy and reliability of the model in recovering member-end P , V_2 , and M_3 .

4.1. Dataset Generation and Data Split

To clarify the data preparation procedure, the numerical dataset used for model development was generated from 2D structural models analyzed using SAP2000. v26.0.0. The dataset contains 2760 graph samples, covering structures with 9 to 100 nodes, with

30 samples generated for each node number. Each sample corresponds to a complete structural graph, where nodes represent structural joints or DOF locations and edges represent structural members. The graph input includes nodal coordinates, boundary condition indicators, nodal load components, member connectivity, member geometry, stiffness-related member parameters, and member load information.

The reference responses were obtained from SAP2000 analyses and then converted into graph-level training data. The reference response data include nodal DOF displacements and member-end force components. Therefore, although the GNN directly predicts nodal DOF displacements, the workflow is connected to engineering structural analysis outputs by recovering member axial forces, shear forces, bending moments, and support reactions from the predicted displacements through structural mechanics relations. In the implemented data files, each sample contains node features, member/edge features, directed edge indices, displacement labels, and member force labels.

For diagnostic training and validation, the generated graph samples were divided at the graph-sample level into training, validation, and testing subsets using a fixed random seed. The split contains 1932 training samples, 414 validation samples, and 414 testing samples, corresponding approximately to a 70/15/15 ratio. The split was performed by complete graph samples rather than by individual nodes, members, or response components. Thus, each structural graph sample appears in only one subset.

This sample-level split reduces direct data leakage between training and evaluation because the same graph sample is not shared across different subsets. Nevertheless, the split should be interpreted as a diagnostic evaluation under the current numerical data-generation setting. Since the samples were generated according to the same modeling rules and within the same general 2D frame/truss structural family, this split does not constitute a strict out-of-distribution test across entirely unseen structural systems. More stringent topology-isolated splits, structural-family-level splits, and validation using independently generated or experimental datasets remain necessary for future work.

4.2. Example Design and Experimental Setup

A planar frame structure is used as the main validation object in this study. The structure consists of beams, columns, and diagonal braces, and each node has three DOFs: horizontal displacement, vertical displacement, and rotation. In the structural graph, nodes correspond to frame joints, and edges correspond to beam-column or bracing members. Node features include nodal coordinates, boundary constraints, and nodal load information, while edge features include member length, orientation, material and section parameters, and loads applied to members. Through this representation, structural geometry, topological connectivity, boundary conditions, member stiffness, and actual member loads are uniformly input into the GNN model (Figure 3).

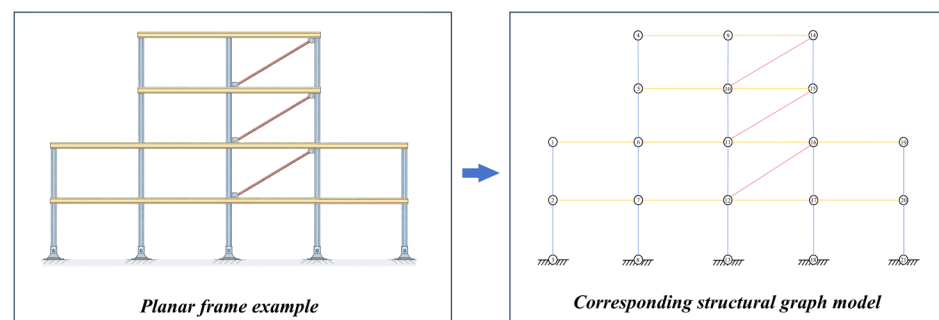


Figure 3. Schematic diagram of the frame-structure example.

For comparison, explicit computational results based on structural mechanics relations are used as reference responses. The reference results include nodal displacements, member-end forces, and support reactions. The GNN model takes the structural graph as input and outputs nodal DOF displacements directly; member axial forces, shear forces, and bending moments are subsequently obtained. All member internal forces are evaluated under a unified member local coordinate system and sign convention to ensure comparability among different model results.

Training used a weighted combination of displacement supervision loss, free DOF equilibrium loss, and global equilibrium loss. The displacement supervision term improves the fitting of reference nodal displacements, while the equilibrium loss terms constrain predictions to satisfy static equilibrium at both the nodal DOF level and the structural-system level. To avoid dimensional effects in the training curves, the normalized equilibrium residual is used as the main monitoring indicator in the training-process analysis. To ensure reproducibility, the network architecture and training hyperparameters used in the main experiment are described in a unified manner. The model settings adopted in the main experiment are listed in Table 4.

Table 4. Main experimental model architecture and training hyperparameters.

Configuration Item	Setting
Network architecture	3 message-passing layers, hidden dimension 128
Message aggregation and activation	Sum aggregation, ReLU activation
Optimizer	Adam training, LBFGS refinement
Learning rate and iterations	Initial learning rate 2×10^{-4} , Adam iterations 2000 steps
Loss weights	Selected according to the initial magnitudes and convergence behavior of the displacement supervision loss, free DOF equilibrium loss, and global equilibrium loss

Because nodal displacement is the direct output of the GNN, this study first evaluates displacement prediction errors on the free DOFs, including the MAE, RMSE, and maximum absolute errors of u_x , u_y , and r_z . On this basis, the equilibrium residual is further used to evaluate the static consistency of the predicted displacement, and the P MAE, V_2 MAE, and M_3 MAE are used to evaluate the accuracy of axial force, shear force, and bending moment recovered from the predicted displacement.

4.3. Evaluation Metrics and Analysis Results for the Actual-Load Frame

The actual load frame example examines the applicability of the proposed method in an engineering static analysis scenario. Unlike simplified examples with only nodal loads, this example inputs loads applied to members as part of the structural graph edge features, enabling external actions on beams, columns, or braces to participate in nodal displacement prediction and subsequent internal force recovery. This setting is closer to practical structural analysis modeling and can also test the model's ability to use actual member load information.

In this example, the proposed method achieves small nodal displacement prediction errors on the free DOFs. The MAE and RMSE of u_x are 0.00148 mm and 0.00234 mm, respectively; those of u_y are 0.000688 mm and 0.000996 mm, respectively; and those of r_z are 0.1295 mrad and 0.2243 mrad, respectively. These results indicate that the nodal displacements directly output by the GNN agree well with the reference solution. On this basis, the normalized equilibrium residual and member internal force errors are further used to evaluate the mechanical consistency and engineering usability of the predictions.

Figure 4 shows the variation in the normalized equilibrium residual during training for the baseline GNN and the mechanically constrained model. The baseline model can learn nodal responses through displacement supervision, but its equilibrium residual remains at a relatively high level after training, indicating that displacement-label fitting alone cannot ensure that the predictions satisfy static equilibrium. In contrast, the equilibrium residual of the mechanically constrained model continuously decreases during training and finally stabilizes at a low level, showing that the free DOF equilibrium residual and global equilibrium residual effectively constrain the mechanical consistency of the GNN output.

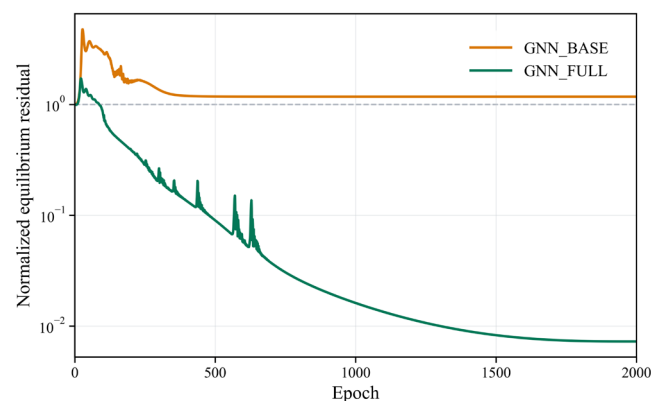


Figure 4. Convergence curves of the normalized equilibrium residual for the baseline GNN and the mechanically constrained model.

The nodal displacements predicted by the full model are substituted into the member internal force recovery process to obtain the axial-force, shear-force, and bending-moment responses of the frame members. To examine the engineering consistency of the recovered results, the internal force diagrams are compared with SAP2000 results. Before comparison, the member local coordinate directions and member-end internal force sign conventions are unified to avoid sign differences caused by different output conventions. This treatment is used only to provide a consistent comparison basis for internal force results and does not change the core analysis workflow of the proposed method, which is based on GNN displacement prediction and mechanical-relation-based recovery.

Figure 5 presents the comparison of member internal force diagrams for the actual-load frame example. In terms of the overall distribution, the full model captures the main force-transfer characteristics of the SAP2000 results well. For loaded beam segments, the trends of the shear-force and bending-moment diagrams are generally consistent with the reference results. For diagonal braces, the axial-force distribution also reflects their main load transfer role. These results indicate that the GNN displacement predictions obtained after mechanically constrained training can be further used for member internal force recovery rather than remaining only at the level of nodal displacement regression.

As Table 5 shows, GNN_BASE achieves some nodal response prediction capability under displacement supervision, but its equilibrium residual and shear force and bending moment errors remain evident. In relative terms, the V_2 and M_3 MAEs of GNN_BASE reach 23.95% and 37.98%, respectively, indicating that displacement fitting alone does not necessarily lead to reliable member force recovery. After the free DOF equilibrium constraint and global equilibrium constraint are introduced, GNN_FULL substantially reduces the equilibrium residual, and the relative MAEs of V_2 and M_3 decrease to 0.40% and 0.74%, respectively. Therefore, in structural static analysis, displacement accuracy alone cannot fully represent engineering reliability. The subsequently recovered member internal forces become mechanically meaningful only when the predicted displacements also satisfy static equilibrium.

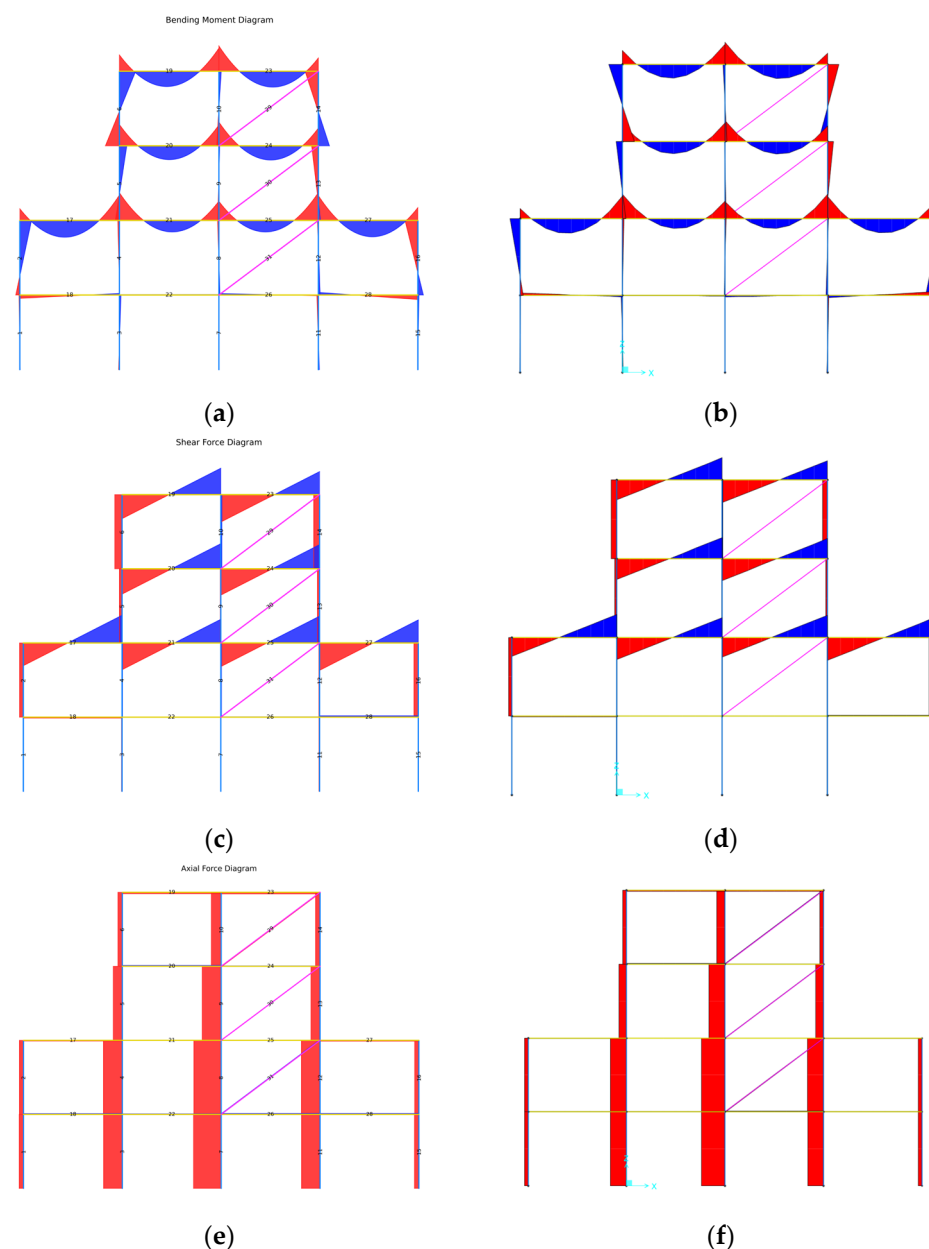


Figure 5. Comparison of internal force diagrams. (a) Proposed method: bending moment; (b) SAP2000: bending moment; (c) Proposed method: shear force; (d) SAP2000: shear force; (e) proposed method: axial force; (f) SAP2000: axial force.

Table 5. Main evaluation metrics of the example.

Loss Term	Equilibrium Residual	P MAE/kN (Relative MAE/%)	V ₂ MAE/kN (Relative MAE/%)	M ₃ MAE/kN·m (Relative MAE/%)
GNN_BASE	2.76×10^{-2}	0.29 (0.49%)	2.98 (23.95%)	5.16 (37.98%)
GNN_FULL	3.90×10^{-9}	0.33 (0.55%)	0.05 (0.40%)	0.10 (0.74%)

Note: The relative MAE is calculated by normalizing the MAE with the mean absolute SAP2000 reference response of the corresponding force component.

The large reduction in the equilibrium residual should be interpreted in relation to the loss-function design rather than as an isolated numerical phenomenon. In the GNN_FULL model, the free DOF equilibrium residual and the global equilibrium residual are explicitly included in the training loss, while GNN_BASE is mainly supervised by displacement errors. Therefore, the equilibrium residual is directly penalized during training

in GNN_FULL, which explains its substantial decrease from the baseline model. This result shows that the added equilibrium constraints improve the static admissibility of the predicted displacement field. Nevertheless, a smaller equilibrium residual does not necessarily imply the same level of improvement in all engineering response quantities. Therefore, displacement errors, member force errors, and equilibrium residuals are reported together to evaluate the model from both data-fitting and mechanics-consistency perspectives.

These numerical results do not imply that the linear elastic static problem is difficult for conventional finite element or stiffness method solvers. Instead, the comparison examines whether GNN-predicted displacements can satisfy the mechanical requirements of structural static analysis. The results indicate that displacement fitting alone is insufficient for engineering static analysis, whereas incorporating free DOF and global equilibrium constraints improves equilibrium consistency and member force recovery.

As a preliminary timing check, the offline training process using the default loss weights for 2000 Adam iterations followed by LBFGS refinement required 334.85 s for the 100-node example. Once the model and data were loaded, a forward prediction for the same example took 6.45 ms on average. The corresponding end-to-end report/inference workflow, including file loading, response evaluation, metric calculation, and Excel export, required 3.31 s. These timings indicate that the trained model provides a fast forward evaluation stage for the tested graph, whereas training remains an offline cost. Broader comparisons over larger structures, different graph topologies, and optimized finite element implementations remain necessary.

In summary, the proposed method predicts nodal displacements from structural graph inputs and obtains engineering responses at the member level through mechanical constraints and internal force recovery. Comparison with SAP2000 results further shows that the internal force diagrams recovered by the full model reflect the main load transfer paths and internal force distribution characteristics, indicating potential for engineering static-analysis applications.

4.4. Ablation Experimental Design and Results Analysis

To further analyze how different components affect structural static analysis results, this study conducts ablation experiments based on the actual load frame example. The ablation experiments examine the effects of mechanical equilibrium constraints, actual member load input, and engineering internal force recovery treatment on result reliability.

Five configurations are selected for comparison. MECH_REF denotes reference results obtained from structural mechanics relations and provides a comparison basis under the mechanical-analysis convention. GNN_BASE is the baseline model trained only with displacement supervision. GNN_FULL is the complete proposed model, incorporating both free DOF equilibrium constraints and global equilibrium constraints. NO_MEMBER_LOAD_INPUT is used to examine the influence of missing structural force information after removing member load input. NO_RECOVERY_CORR is used to examine the influence of omitting load correction and local-coordinate sign unification during internal force recovery on member internal force results. The two diagnostic groups serve different purposes: NO_MEMBER_LOAD_INPUT clarifies whether actual member loads should enter the graph input, whereas NO_RECOVERY_CORR demonstrates the necessity of the internal force recovery process rather than evaluating nodal displacement prediction accuracy. The group settings are listed in Table 6.

Table 6. Ablation groups and their physical meanings.

Group	Meaning
MECH_REF	Reference results based on structural mechanics relations
GNN_BASE	Baseline model trained with displacement supervision only
GNN_FULL	Full model with displacement supervision and free DOF equilibrium constraints
NO_MEMBER_LOAD_INPUT	Removes member load input to test the necessity of actual member load information
NO_RECOVERY_CORR	Removes internal force recovery correction to test the necessity of engineering recovery treatment

The ablation results are shown in Table 7, where P , V_2 , and M_3 denote axial force, shear force, and bending moment in the member local coordinate system, respectively.

Table 7. Key ablation results of the example.

Group	Equilibrium Residual	P MAE/kN	V_2 MAE/kN	M_3 MAE/kN·m
MECH_REF	6.01×10^{-32}	0.33	0.05	0.10
GNN_BASE	2.76×10^{-2}	0.29	2.98	5.16
GNN_FULL	3.90×10^{-9}	0.33	0.05	0.10
NO_MEMBER_LOAD_INPUT	2.00×10^{-2}	31.22	1.20	2.05
NO_RECOVERY_CORR	6.01×10^{-32}	0.33	6.24	4.82

The comparison between GNN_BASE and GNN_FULL shows that displacement supervision alone cannot guarantee static equilibrium. The equilibrium residual of GNN_BASE is 2.76×10^{-2} , which is clearly higher than that of the full model, and its V_2 MAE and M_3 MAE reach 2.98 and 5.16, respectively. By contrast, the equilibrium residual of GNN_FULL decreases to 3.90×10^{-9} , and the shear-force and bending-moment errors are also close to the mechanics-based reference results. This result shows that, after the free DOF equilibrium residual and global equilibrium residual are introduced into the training objective, the GNN-predicted displacement is not only numerically close to the reference displacement, but also more consistent with structural static equilibrium requirements. This comparison further confirms that the role of the proposed mechanical constraints is not to replace the stiffness equation, but to prevent a displacement prediction model from producing mechanically inconsistent structural analysis outputs.

Removing member load input causes a marked error increase, especially with the P MAE reaching 31.22. This phenomenon indicates that, when loads applied to members are not included as part of the structural graph input, the model or mechanical workflow receives incomplete loading information and cannot accurately reflect the effect of actual loads on member axial forces and the overall loading state. For real frame structures, external actions are not always concentrated at nodes. Load information on beams, columns, or braces must enter the analysis workflow as structural graph edge features. Even when structural topology and stiffness information are complete, the model may still predict based on incomplete external-action information.

Omitting the recovery correction leaves the equilibrium residual close to zero, but its V_2 MAE and M_3 MAE increase to 6.24 and 4.82, respectively. This indicates that correct nodal displacement and equilibrium relations do not necessarily guarantee correct member internal force diagrams. Shear force and bending moment are sensitive to member loading, local coordinate direction, and member-end internal force sign convention. If the internal force recovery stage lacks the corresponding corrections, the recovered bending-shear internal forces may still show evident deviations even if the displacement field satisfies

global static equilibrium. Therefore, engineering internal force recovery is an important link between GNN displacement predictions and member-level analysis results.

Combining Table 5 and Figure 6, the full proposed model shows good mechanical consistency for the equilibrium residual, shear force error, and bending moment error. The GNN provides displacement response prediction capability for structural graph topology, mechanical equilibrium constraints improve the static credibility of the predicted displacement, and actual member load input together with engineering internal force recovery ensures that the model output can be further transformed into member-level P , V_2 , and M_3 results. The ablation experiments show that the effectiveness of the proposed method does not arise from a single module; it comes from graph neural network prediction, mechanical-equation constraints, and member internal force recovery working together.

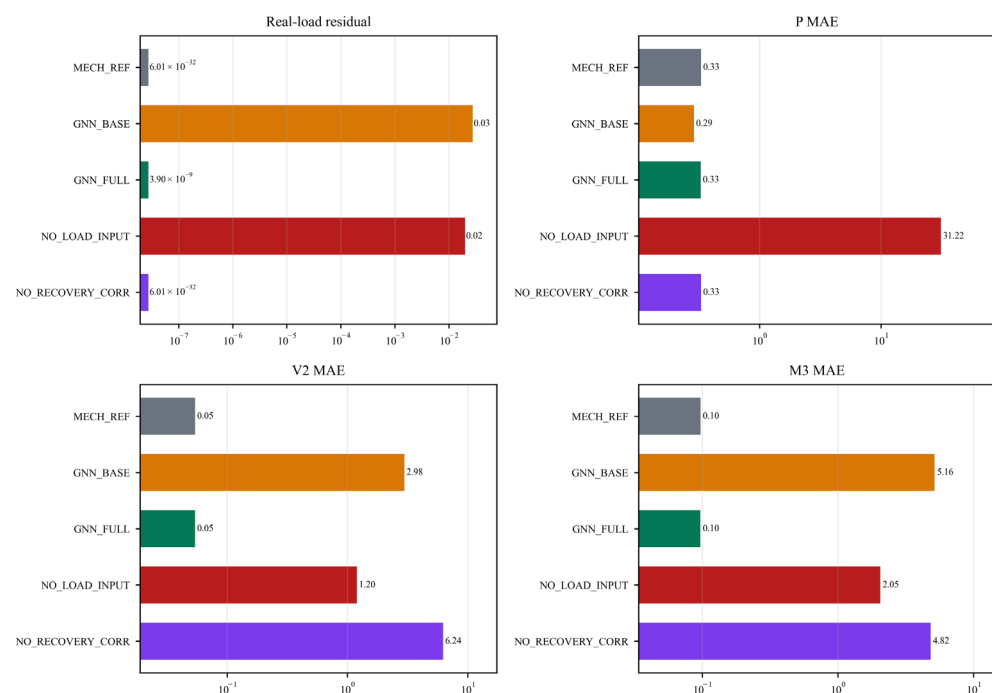


Figure 6. Comparison of equilibrium residuals and member internal force errors under different model configurations.

4.5. Extended Validation Examples

The preceding results are mainly based on a planar frame example with actual member loads. To further examine the applicability of the proposed method under different structural topologies and load input forms, this study adds two extended validation examples beyond the main example. These examples provide preliminary evidence that the proposed workflow can handle different structural forms, including structural graph representation, boundary condition treatment, displacement prediction, equilibrium constraints, and internal force recovery. However, they serve as supplemental validation rather than a comprehensive test of cross-topology generalization.

Extended Example 1 is a multi-story braced planar frame structure whose number of stories, number of spans, and brace arrangement differ from those of the main example, and is used to examine model applicability under a more complex frame topology. Extended Example 2 is a low-rise braced frame structure, and is used to further test displacement prediction and internal force recovery for a smaller frame with local bracing members. Both extended examples use the same node features, edge features, boundary mask treatment, and mechanical-constraint loss function as the main experiment, and explicit structural analysis results are used as reference solutions.

Figure 7 presents the structural schematic and internal force recovery comparison for Extended Example 1. Figure 7a shows the structural schematic; Figure 7b–d show the bending-moment, shear-force, and axial-force diagrams recovered from the GNN-predicted displacements, respectively; and Figure 7e–g show the reference bending-moment, shear-force, and axial-force diagrams, respectively. The results show that, for this multi-story braced frame, the predictions can reflect the distribution characteristics of different internal forces along beams, columns, and braces, and the overall trends of bending moment, shear force, and axial force are consistent with the reference results.

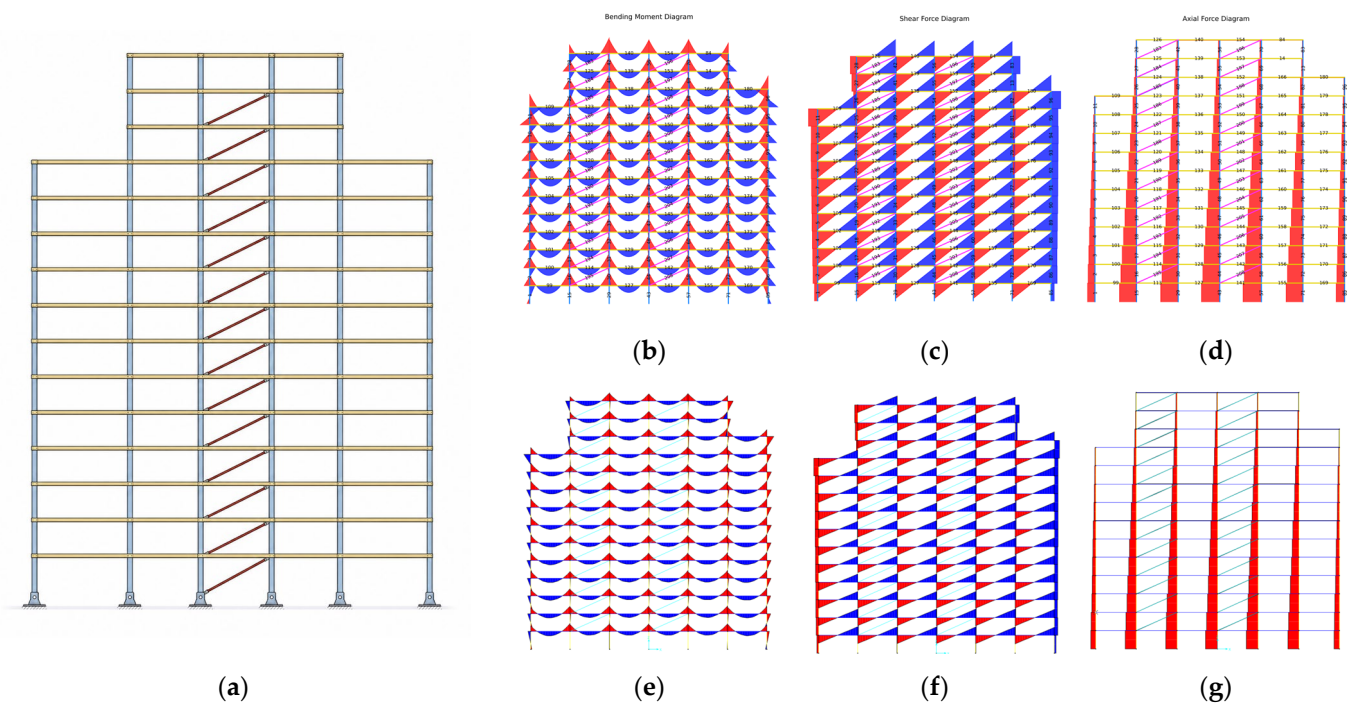


Figure 7. Structural schematic and internal force recovery comparison for Extended Example 1. (a) Structural schematic; (b) predicted bending moment diagram; (c) predicted shear force diagram; (d) predicted axial force diagram; (e) reference bending moment diagram; (f) reference shear force diagram; (g) reference axial force diagram.

Figure 8 presents the structural schematic and internal force recovery comparison for Extended Example 2. Compared with Extended Example 1, this example is smaller in scale but contains different support constraints and local diagonal braces. As shown in Figure 8, the predicted and reference internal force diagrams exhibit generally consistent distribution trends in the main load-bearing members, indicating that the proposed method can maintain good internal force recovery capability in frame structures of different scales and connection forms.

To provide a further quantitative evaluation, this study evaluates nodal displacement errors on the free DOFs, normalized free DOF equilibrium residuals, and member internal force errors for the extended examples. Error statistics are calculated only for free DOFs to avoid dilution of displacement-error evaluation by constrained support displacements. The statistical results are listed in Table 8.

As shown in Table 8, both extended examples maintain small prediction errors in the u_x , u_y , and r_z DOFs, and their normalized equilibrium residuals are 4.71×10^{-9} and 1.28×10^{-15} , respectively, indicating that the predicted displacements satisfy free DOF static equilibrium well. Meanwhile, the mean absolute errors of member axial force, shear force, and bending moment in both extended examples remain small, showing good agreement between the internal forces recovered from the predicted displacements and the reference

results. Together with Figures 7 and 8, these results indicate that the proposed method can be applied beyond the main example to braced frame structures with different topologies and scales, preliminarily demonstrating the applicability of the structural graph input, mechanically constrained training, and internal force recovery workflow.

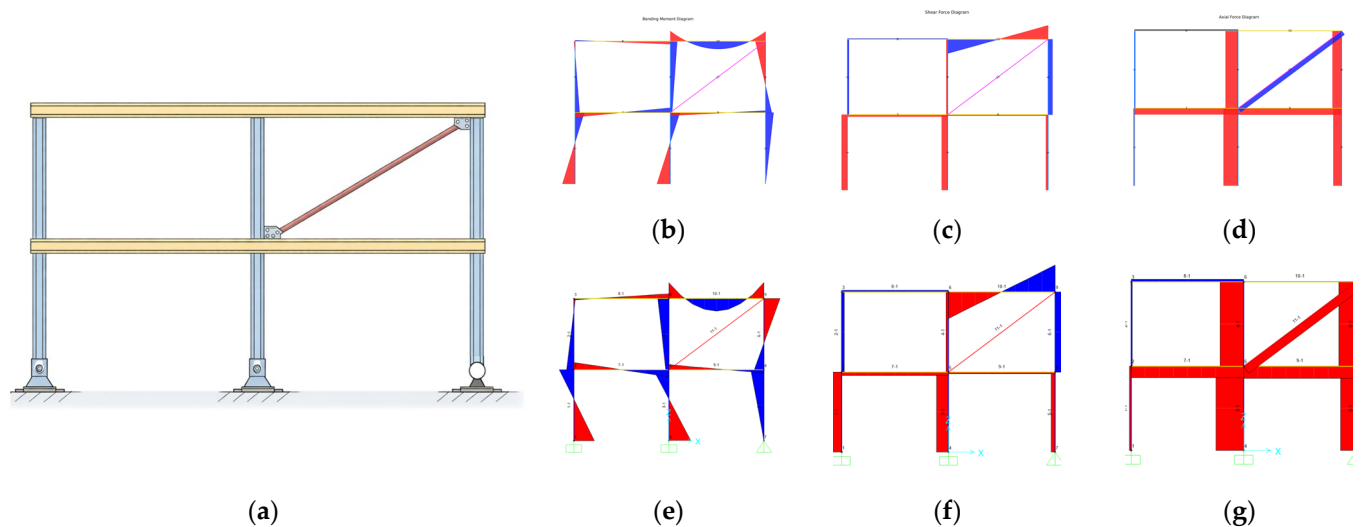


Figure 8. Structural schematic and internal force recovery comparison for Extended Example 2. (a) Structural schematic; (b) predicted bending moment diagram; (c) predicted shear force diagram; (d) predicted axial force diagram; (e) reference bending moment diagram; (f) reference shear force diagram; (g) reference axial force diagram.

Table 8. Error statistics for the extended validation examples.

Example	u_x MAE /mm	u_y MAE /mm	r_z MAE /mrad	Equilibrium Residual	P MAE /kN	V_2 MAE /kN	M_3 MAE /kN·m
Extended example 1	0.123	0.209	0.0154	4.71×10^{-9}	0.142	0.0515	0.104
Extended example 2	0.0486	0.00025	0.00585	1.28×10^{-15}	0.0980	0.0643	0.0989

5. Discussion

5.1. Role of Graph Representation in Structural Static Analysis

This study does not simplify the GNN into a nodal displacement regression tool. Instead, it examines its feasibility for engineering applications in structural static analysis. The numerical results show that, after a planar frame structure is represented as graph data consisting of nodes and members, the GNN can learn nodal displacement responses under the combined influence of structural topology, boundary constraints, member stiffness, and external actions. This result indicates that graph neural networks have good formal compatibility with structural systems such as trusses and frames.

Compared with ordinary neural networks using fixed-dimensional inputs, GNNs have the advantage of directly handling the number of nodes, member connectivity, and local neighborhood relationships. The loading state of a structural system is not determined independently by a single node or member, but is jointly formed by nodal connections, member stiffness, support conditions, and external loads. The message-passing mechanism can transmit information between nodes along member connections, allowing nodal hidden vectors to progressively incorporate neighboring members and broader structural topology.

Therefore, using GNNs for structural static analysis has a clear structural mechanics basis rather than relying only on data fitting.

The proposed method is intended to complement conventional finite element and stiffness method solvers by providing a mechanically interpretable graph learning workflow. In this workflow, the graph representation preserves node-member topology and member load information, equilibrium residuals introduce static mechanical constraints during training, and the recovery step based on structural mechanics converts predicted nodal displacements into support reactions and member internal forces, including axial forces, shear forces, and bending moments. These features allow the framework to extend beyond nodal displacement prediction and recover engineering response quantities such as support reactions and member internal forces. Therefore, the framework should be viewed as a complementary graph learning tool for structural response prediction and interpretation, while final design verification remains the role of validated structural analysis procedures. Although the preliminary timing check in Section 4.3 shows a fast forward-evaluation stage for the tested graph after offline training, broader efficiency assessment is still needed over larger structural scales, different graph topologies, more load cases, and optimized finite element or stiffness method implementations. Graph-based learning has also shown potential in multiscale simulations, where microstructure-based graph neural networks have been used to accelerate simulations involving complex material behavior [46].

The framework may therefore support preliminary scheme comparison, response screening over multiple load cases, and research-oriented surrogate analysis of node-member structural systems. These applications should remain auxiliary to validated finite element or stiffness method analyses used for final design verification.

5.2. Influence of Mechanical Constraints on Prediction Credibility

The ablation experiments show that GNN_BASE, trained only with displacement supervision, can learn nodal displacement responses, but its equilibrium residual and bending-shear internal force errors remain relatively large. Thus, a small displacement error does not necessarily mean that the structural response satisfies static equilibrium, nor does it guarantee reliable member internal forces recovered from displacement. For structural static analysis, model outputs must simultaneously satisfy data-fitting and mechanical-consistency requirements. Similar concerns have also been reported in learning-based elastoplastic modeling, where recurrent neural-network models for elasto-plastic solids require self-consistency to avoid mechanically inconsistent path-dependent responses [47].

This study discretizes the structural static equilibrium equation at the nodal DOF level and constructs equilibrium residuals on the free DOFs. This treatment avoids interference from support reactions at constrained DOFs in the loss function and applies mechanical constraints to the free DOFs that should satisfy equilibrium conditions. The results show that the full model reduces both the equilibrium residual and the member bending-shear internal force errors compared with the purely supervised model. This comparison indicates that the free DOF equilibrium constraint improves the mechanical credibility of GNN-predicted displacements.

Training need not rely entirely on large displacement label datasets once mechanical equations enter the loss function. Displacement labels can improve training stability and prediction accuracy, while equilibrium residuals provide training signals from structural mechanics equations. In engineering structural analysis, high-quality labels often rely on finite element software or experimental results and are costly to obtain. Introducing mechanical constraints helps reduce dependence on large-scale displacement labels and makes the training process closer to the structural analysis problem itself.

5.3. Significance of Actual Member-Load Input and Engineering Internal-Force Recovery

Actual member load input is important for applying the proposed method to engineering problems. External actions in real frame structures do not always appear as nodal forces; beams, columns, or braces may be subjected to distributed loads, concentrated loads, or other parameterized loads. If the model input contains only nodal loads, the external-action information represented by the structural graph is incomplete. In the ablation results, removing member load input leads to a clear increase in internal force error, indicating that actual member load information must enter the structural graph representation and displacement prediction process.

At the same time, the member internal force recovery process is critical for interpreting the results. The direct output of the GNN is nodal displacement, whereas engineering analysis usually requires member internal force results such as P , V_2 , and M_3 . This study converts the predicted displacements into member-level engineering mechanical quantities through member local-coordinate transformation, local stiffness relations, and internal force recovery correction. The ablation results show that, when internal force recovery correction is removed, shear-force and bending-moment errors increase markedly even if the equilibrium residual remains close to the reference state. This indicates that correct nodal displacement does not automatically guarantee correct internal force diagrams; the internal force recovery process must still follow structural mechanics relations and engineering output conventions. This is consistent with recent learning-based elastoplastic studies showing that neural-network mechanics models should preserve energy-based or mechanics-based formulations when extended to nonlinear material behavior [48].

The comparison with SAP2000 internal force diagrams further shows that the full model can capture the axial-force, shear-force, and bending-moment distribution characteristics of the main members. Engineering software results are used here as a consistency reference to examine whether the model output conforms to common engineering analysis conventions, rather than as a component of the proposed method itself. This comparison demonstrates that internal force recovery based on predicted displacements can extend the GNN output to the member-response level required for engineering static analysis.

5.4. Limitations and Future Work

The present study focuses on 2D planar truss and frame structures. The structural graph representation, free DOF equilibrium constraints, and member internal force recovery process are all based on 2D linear elastic static analysis. Practical 3D spatial structures involve more complex nodal DOFs, local coordinate transformations, load expressions, and internal force recovery forms. Therefore, future research will further extend the proposed framework, as shown in Figure 9, so that it can be applied to 3D frames and spatial truss structures and validated under more structural scales and load cases. After the 3D framework is established, future work may further investigate material nonlinearity, geometric nonlinearity, dynamic response, impact, and vibration problems. Related studies on the impact response of stiffened bottom tank structures and the nonlinear vibration of composite stiffened NiTiNOL-steel wire ropes and panels further highlight the importance of nonlinear and dynamic extensions for complex stiffened structural systems [49,50]. These extensions are expected to be more meaningful for 3D structural systems, where spatial load transfer, member coupling, torsion, biaxial bending, and nonlinear deformation mechanisms are more prominent than in the present 2D linear elastic setting.

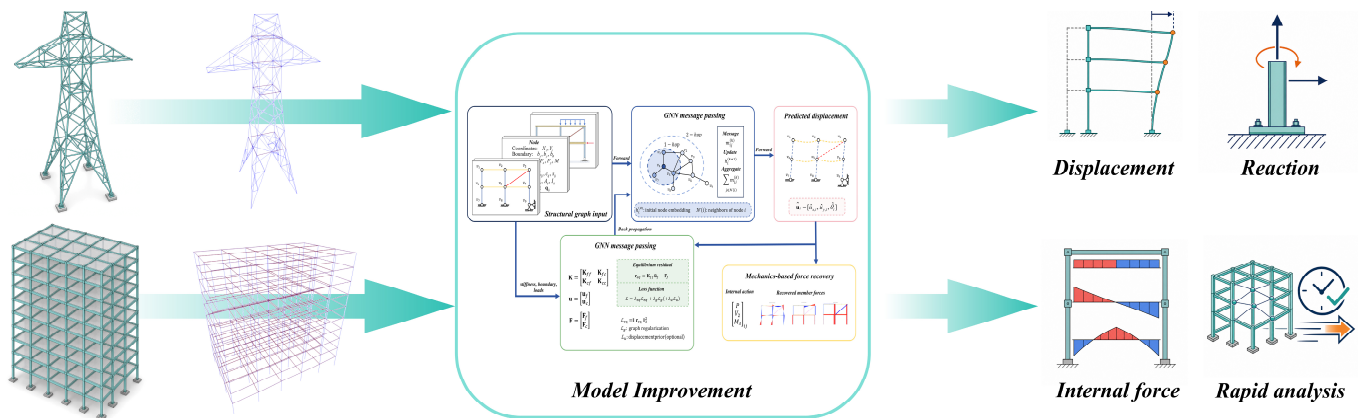


Figure 9. Engineering application positioning and future extension directions.

In addition, the present study uses fixed weighting coefficients for the displacement-supervision loss, free DOF equilibrium loss, and global equilibrium loss. To examine their influence, a preliminary loss-weight sensitivity check was conducted on the 100-node example by varying the free DOF equilibrium weight, the displacement-supervision weight schedule, and the global equilibrium weight around the default setting. The results show that removing or substantially weakening the free DOF equilibrium term increases the equilibrium residual loss, while removing the global equilibrium term mainly increases the global equilibrium loss. Moderate changes in the supervision and global equilibrium weights led to similar member force errors in this controlled example. Therefore, the default weights were used as a balanced setting to maintain displacement fitting, free DOF equilibrium, and global equilibrium during training. Nevertheless, this check remains limited to one representative example, and a more systematic sensitivity study over different structural scales, load cases, and graph topologies is still needed to provide general guidance for selecting λ_u , λ_{eq} , and λ_g .

The extended examples in this study are intended to provide preliminary applicability checks under several 2D linear elastic static-analysis settings, rather than a comprehensive validation over all possible structural configurations.

6. Conclusions

This study proposes a graph neural network method incorporating mechanical constraints for response surrogate modeling in structural static analysis. The method uniformly represents structural nodes, member connectivity, boundary conditions, material and section parameters, nodal loads, and actual member loads as a structural graph, enabling the GNN to learn the mapping between the overall loading state and nodal displacement response on the node-member topology. Compared with conventional fixed-dimensional input models, this graph representation more naturally describes the irregular connection characteristics of planar truss and frame structures and provides a unified data basis for applying GNNs to structural static analysis.

During training, the method explicitly introduces the structural static equilibrium relation into the loss function. DOF-level equilibrium residuals and global equilibrium residuals jointly constrain displacement prediction. This strategy prevents training from relying entirely on large displacement label datasets and allows structural mechanical equations to provide effective supervision signals. The numerical results show that the model trained only with displacement supervision can fit nodal responses, but its equilibrium consistency is weak and its shear force and bending moment errors remain large. After mechanical constraints are introduced, the model improves both equilibrium residuals and internal force recovery accuracy.

At the engineering-output level, this study further recovers member-end forces and support reactions from the predicted nodal displacements and, by considering actual member loads and internal force sign conventions in the local coordinate system, obtains structural static-analysis results such as P , V_2 , and M_3 . The frame example and ablation experiments show that member-load input, equilibrium constraints, and internal force recovery correction all have significant effects on the final analysis accuracy. Removing member-load input weakens the model's ability to represent the actual loading state, while removing internal force recovery correction leads to evident deviations in shear-force and bending-moment results. These findings indicate that each component of the proposed method has a clear mechanical role and engineering necessity.

In conclusion, this study does not treat the GNN as a simple displacement regression tool. It constructs a graph neural network workflow for engineering static analysis, in which the GNN predicts nodal displacements, mechanical equations constrain the predictions to satisfy static equilibrium, and structural mechanics relations recover member internal forces. This improves the mechanical consistency, interpretability, and engineering applicability of the model output. The present study establishes a controlled 2D linear elastic static analysis framework for mechanically constrained graph learning of node-member structural systems. Future work will extend the framework to 3D spatial structures and then investigate nonlinear material behavior, geometric nonlinearity, dynamic response, impact, and vibration problems.

Author Contributions: Conceptualization, K.L. and P.H.; methodology, K.L. and P.H.; software, K.L.; validation, K.L., H.J. and Q.F.; formal analysis, K.L.; investigation, K.L. and P.H.; resources, P.H.; data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, P.H., H.J., Q.F., W.Z. and Y.Z.; visualization, K.L.; supervision, P.H.; project administration, P.H.; funding acquisition, P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the China Postdoctoral Science Foundation, grant number 2022M711962.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to institutional restrictions and the ongoing use of the dataset in related research.

Conflicts of Interest: Author Wen Zheng was employed by the company Fujian Huichuan IoT Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GNN	Graph neural network
DOF	Degree of freedom
FEM	Finite element method
FEA	Finite element analysis
PINN	Physics-informed neural network
MAE	Mean absolute error
GNN_BASE	Baseline GNN model
GNN_FULL	Full GNN model with mechanical constraints
MECH_REF	Mechanics-based reference result
NO_MEMBER_LOAD_INPUT	Model without member-load input
NO_RECOVERY_CORR	Model without internal force recovery correction

References

1. Salehi, H.; Burgueño, R. Emerging artificial intelligence methods in structural engineering. *Eng. Struct.* **2018**, *171*, 170–189. [\[CrossRef\]](#)
2. Thai, H.T. Machine learning for structural engineering: A state-of-the-art review. *Structures* **2022**, *38*, 448–491. [\[CrossRef\]](#)
3. Tapeh, A.T.G.; Naser, M.Z. Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. *Arch. Comput. Methods Eng.* **2022**, *30*, 115–159. [\[CrossRef\]](#)
4. Sun, H.; Burton, H.V.; Huang, H. Machine learning applications for building structural design and performance assessment: State-of-the-art review. *J. Build. Eng.* **2020**, *33*, 101816. [\[CrossRef\]](#)
5. Jayasinghe, S.; Mahmoodian, M.; Alavi, A.; Sidiq, A.; Shahrivar, F.; Sun, Z.; Thangarajah, J.; Setunge, S. A review on the applications of artificial neural network techniques for accelerating finite element analysis in the civil engineering domain. *Comput. Struct.* **2025**, *310*, 107698. [\[CrossRef\]](#)
6. Liang, L.; Liu, M.; Martin, C.F.; Sun, W. A deep learning approach to estimate stress distribution: A fast and accurate surrogate of finite-element analysis. *J. R. Soc. Interface* **2018**, *15*, 20170844. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Bolandi, H.; Li, X.; Salem, T.; Boddetti, V.N.; Lajnef, N. Bridging finite element and deep learning: High-resolution stress distribution prediction in structural components. *Front. Struct. Civ. Eng.* **2022**, *16*, 1365–1377. [\[CrossRef\]](#)
8. Bolandi, H.; Li, X.; Salem, T.; Boddetti, V.N.; Lajnef, N. Deep learning paradigm for prediction of stress distribution in damaged structural components with stress concentrations. *Adv. Eng. Softw.* **2022**, *173*, 103240. [\[CrossRef\]](#)
9. Triantafyllou, G.; Kaloizoumis, P.G.; Dimas, G.; Iakovidis, D.K. DeepFEA: Deep learning for prediction of transient finite element analysis solutions. *Expert Syst. Appl.* **2024**, *269*, 126343. [\[CrossRef\]](#)
10. Le-Duc, T.; Nguyen-Xuan, H.; Lee, J. A finite-element-informed neural network for parametric simulation in structural mechanics. *Finite Elem. Anal. Des.* **2023**, *217*, 103904. [\[CrossRef\]](#)
11. Zhang, N.; Xu, K.; Yin, Z.Y.; Li, K.K.; Jin, Y.F. Finite element-integrated neural network framework for elastic and elastoplastic solids. *Comput. Methods Appl. Mech. Eng.* **2024**, *433*, 117474. [\[CrossRef\]](#)
12. Ghavamian, F.; Simone, A. Accelerating multiscale finite element simulations of history-dependent materials using a recurrent neural network. *Comput. Methods Appl. Mech. Eng.* **2019**, *357*, 112594. [\[CrossRef\]](#)
13. Hoffer, J.G.; Geiger, B.C.; Ofner, P.; Kern, R. Mesh-Free Surrogate Models for Structural Mechanic FEM Simulation: A Comparative Study of Approaches. *Appl. Sci.* **2021**, *11*, 9411. [\[CrossRef\]](#)
14. Hu, H.; Qi, L.; Chao, X. Physics-informed Neural Networks (PINN) for computational solid mechanics: Numerical frameworks and applications. *Thin-Walled Struct.* **2024**, *205*, 112495. [\[CrossRef\]](#)
15. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [\[CrossRef\]](#)
16. Cuomo, S.; Cola, V.S.D.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What's Next. *J. Sci. Comput.* **2022**, *92*, 88. [\[CrossRef\]](#)
17. Haghighat, E.; Raissi, M.; Moure, A.; Gómez, H.; Juanes, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Eng.* **2021**, *379*, 113741. [\[CrossRef\]](#)
18. Wang, L.; Liu, G.; Wang, G.; Zhang, K. M-PINN: A mesh-based physics-informed neural network for linear elastic problems in solid mechanics. *Int. J. Numer. Methods Eng.* **2024**, *125*, e7444. [\[CrossRef\]](#)
19. Wang, J.; Mo, Y.L.; Izzuddin, B.A.; Kim, C.W. Exact Dirichlet boundary Physics-informed Neural Network EPINN for solid mechanics. *Comput. Methods Appl. Mech. Eng.* **2023**, *414*, 116184. [\[CrossRef\]](#)
20. Diao, Y.; Yang, J.; Zhang, Y.; Zhang, D.; Du, Y. Solving multi-material problems in solid mechanics using physics-informed neural networks based on domain decomposition technology. *Comput. Methods Appl. Mech. Eng.* **2023**, *413*, 116120. [\[CrossRef\]](#)
21. Bai, J.; Rabczuk, T.; Gupta, A.; Alzubaidi, L.; Gu, Y. A physics-informed neural network technique based on a modified loss function for computational 2D and 3D solid mechanics. *Comput. Mech.* **2022**, *71*, 543–562. [\[CrossRef\]](#)
22. Abueidda, D.; Korić, S.; Guleryuz, E.; Sobh, N. Enhanced physics-informed neural networks for hyperelasticity. *Int. J. Numer. Methods Eng.* **2022**, *124*, 1585–1601. [\[CrossRef\]](#)
23. Haghighat, E.; Bekar, A.C.; Madenci, E.; Juanes, R. A nonlocal physics-informed deep learning framework using the peridynamic differential operator. *Comput. Methods Appl. Mech. Eng.* **2021**, *385*, 114012. [\[CrossRef\]](#)
24. Fuhg, J.N.; Bouklas, N. The mixed Deep Energy Method for resolving concentration features in finite strain hyperelasticity. *J. Comput. Phys.* **2021**, *451*, 110839. [\[CrossRef\]](#)
25. Wang, S.; Wang, H.; Perdikaris, P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Sci. Adv.* **2021**, *7*, eabi8605. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Rao, C.; Sun, H.; Liu, Y. Physics-Informed Deep Learning for Computational Elastodynamics without Labeled Data. *J. Eng. Mech.* **2021**, *147*, 04021043. [\[CrossRef\]](#)
27. Song, L.-H.; Wang, C.; Fan, J.S.; Lu, H.M. Elastic structural analysis based on graph neural network without labeled data. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *38*, 1307–1323. [\[CrossRef\]](#)

28. Chou, Y.T.; Chang, W.-T.; Jean, J.; Chang, K.-H.; Huang, Y.N.; Chen, C.S. StructGNN: An efficient graph neural network framework for static structural analysis. *Comput. Struct.* **2024**, *299*, 107385. [\[CrossRef\]](#)
29. Parisi, F.; Ruggieri, S.; Lovreglio, R.; Fanti, M.P.; Uva, G. On the use of mechanics-informed models to structural engineering systems: Application of graph neural networks for structural analysis. *Structures* **2023**, *59*, 105712. [\[CrossRef\]](#)
30. 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. [\[CrossRef\]](#)
31. Jiang, C.; Chen, N.-Z. Graph Neural Networks (GNNs) based accelerated numerical simulation. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106370. [\[CrossRef\]](#)
32. Maurizi, M.; Gao, C.; Berto, F. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Sci. Rep.* **2022**, *12*, 21834. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Cai, Y.; Jelovica, J. Efficient graph representation in graph neural networks for stress predictions in stiffened panels. *Thin-Walled Struct.* **2024**, *203*, 112157. [\[CrossRef\]](#)
34. Feng, B.; Zhou, X. Energy-informed graph transformer model for solid mechanical analyses. *Commun. Nonlinear Sci. Numer. Simul.* **2024**, *137*, 108103. [\[CrossRef\]](#)
35. Feng, B.; Zhou, X. The novel graph transformer-based surrogate model for learning physical systems. *Comput. Methods Appl. Mech. Eng.* **2024**, *432*, 117410. [\[CrossRef\]](#)
36. Gao, H.; Zahr, M.J.; Wang, J. Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* **2022**, *390*, 114502. [\[CrossRef\]](#)
37. Dalton, D.R.; Husmeier, D.; Gao, H. Physics-informed graph neural network emulation of soft-tissue mechanics. *Comput. Methods Appl. Mech. Eng.* **2023**, *417*, 116351. [\[CrossRef\]](#)
38. Taç, V.; Costabal, F.S.; Tepole, A.B. Data-driven tissue mechanics with polyconvex neural ordinary differential equations. *Comput. Methods Appl. Mech. Eng.* **2022**, *398*, 115248. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Flaschel, M.; Kumar, S.; Lorenzis, L.D. Unsupervised discovery of interpretable hyperelastic constitutive laws. *Comput. Methods Appl. Mech. Eng.* **2021**, *381*, 113852. [\[CrossRef\]](#)
40. Janssen, J.A.; Haikal, G.; DeCarlo, E.C.; Hartnett, M.J.; Kirby, M. A Physics-Informed General Convolutional Network for the Computational Modeling of Materials with Damage. *J. Comput. Inf. Sci. Eng.* **2023**, *24*, 111002. [\[CrossRef\]](#)
41. Afshari, S.S.; Enayatollahi, F.; Xu, X.; Liang, X. Machine learning-based methods in structural reliability analysis: A review. *Reliab. Eng. Syst. Saf.* **2021**, *219*, 108223. [\[CrossRef\]](#)
42. Jeong, H.; Bai, J.; Batuwatta-Gamage, C.; Rathnayaka, C.; Zhou, Y.; Gu, Y. A Physics-Informed Neural Network-based Topology Optimization (PINTO) framework for structural optimization. *Eng. Struct.* **2022**, *278*, 115484. [\[CrossRef\]](#)
43. Ramu, P.; Thananjayan, P.; Acar, E.; Bayrak, G.; Park, J.W.; Lee, I. A survey of machine learning techniques in structural and multidisciplinary optimization. *Struct. Multidiscip. Optim.* **2022**, *65*, 266. [\[CrossRef\]](#)
44. Tian, X.; Wang, J.; Kim, C.W.; Deng, X.; Zhu, Y.-J. Automated machine learning exact dirichlet boundary physics-informed neural networks for solid mechanics. *Eng. Struct.* **2025**, *330*, 119884. [\[CrossRef\]](#)
45. Thawon, I.; Vo, D.; Bui, T.Q.; Rattanamongkhonkun, K.; Chamroon, C.; Tippayawong, N.; Mona, Y.; Wanison, R.; Suttakul, P. Physics-Informed Neural Networks: Current Progress and Challenges in Computational Solid and Structural Mechanics. *Comput. Model. Eng. Sci.* **2026**, *146*, 1. [\[CrossRef\]](#)
46. Storm, J.; Rocha, I.B.C.M.; van der Meer, F.P. A microstructure-based graph neural network for accelerating multiscale simulations. *Comput. Methods Appl. Mech. Eng.* **2024**, *427*, 117001. [\[CrossRef\]](#)
47. Bonatti, C.; Mohr, D. On the importance of self-consistency in recurrent neural network models representing elasto-plastic solids. *J. Mech. Phys. Solids* **2022**, *158*, 104697. [\[CrossRef\]](#)
48. He, J.; Abueidda, D.; Abu Al-Rub, R.; Koric, S.; Jasiuk, I. A deep learning energy-based method for classical elastoplasticity. *Int. J. Plast.* **2023**, *162*, 103531. [\[CrossRef\]](#)
49. Prabowo, A.R.; Sohn, J.M.; Putranto, T. Crashworthiness performance of stiffened bottom tank structure subjected to impact loading conditions: Ship-rock interaction. *Curved Layer. Struct.* **2019**, *6*, 245–258. [\[CrossRef\]](#)
50. Putranto, T.; Yulianto, T.; Sujatanti, S.H.; Setyawan, D.; Zakki, A.F.; Muis Alie, M.Z.; Wibowo, W. Numerical Analysis of Composite Stiffened NiTiNOL-Steel Wire Ropes and Panels Undergoing Nonlinear Vibrations. *Modelling* **2025**, *6*, 77. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.